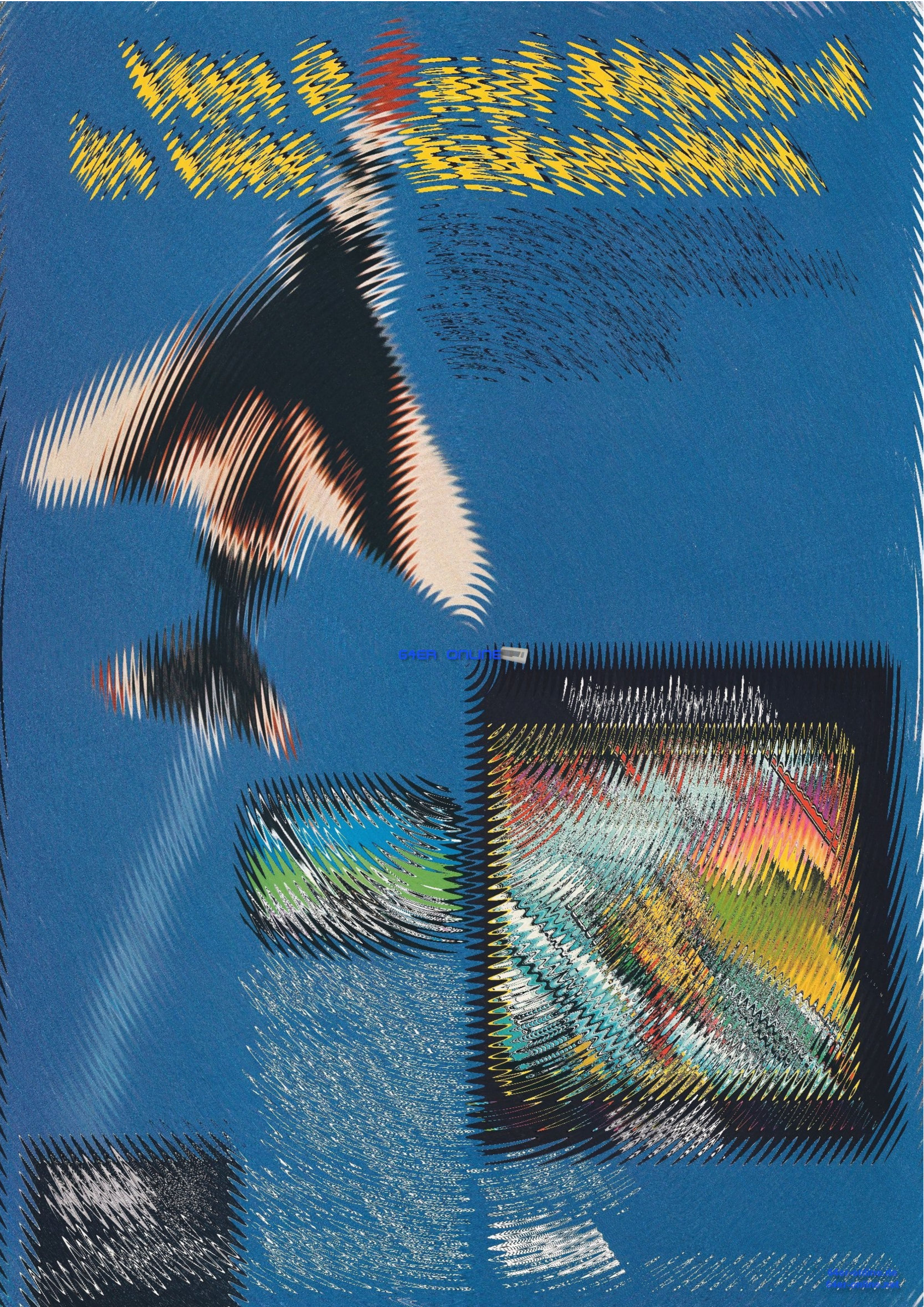


JULI 1985
OS 55,–/Sfr. 6,50
Lit 5.500/hfl 8,–/dkr 30,–
DM 6,50
64'er
7/85
DAS MAGAZIN FÜR COMPUTER-FANS
Eine Markt & Technik Publikation
Datenfern-übertragung
★ Alle Modems und Akustikkoppler
★ Terminalprogramm zum Abtippen
PC 128 und 1571
★ Test: Die neue Floppy
★ Das Betriebssystem CP/M
Super-Assembler zum Abtippen
Vergleichstest Datenbanken
Promal im Test
Die ideale Sprache
Tips & Tricks für C 64, C 16 und VC 20
EPROMs: Grundlagen und Programmiergeräte ★ So schrumpfen Programme ★ Bauanleitung: IEEE-Interface, Betriebssysteme zum Umschalten ★ Haushaltsbuch zum Abtippen


64ER ONLINE



Aktuell

Hannover-Messe 8
Neue Produkte 12

Hardware-Test

Erster ausführlicher Test (Teil 2): C 128 und 1571
★ Test: Die neue Floppy
★ Das Betriebssystem CP/M 17

Drucker

Die Alternativen 24
Test: Brother EP 44 27
Marktübersicht: Drucker 28

Datenfernübertragung

Mailbox für Anfänger 30
Alle Modems und Akustikkoppler
Marktübersicht 32

EPROMs

Grundlagen und Programmiergeräte
Wie sag ich es meinem EPROM? 36
Test: Programmiergeräte
Frisch gebrannt ist halb gespeichert 39

Hardware

Bauanleitung: Betriebssystem zum Umschalten 41
Bauanleitung: IEEE-Interface 44

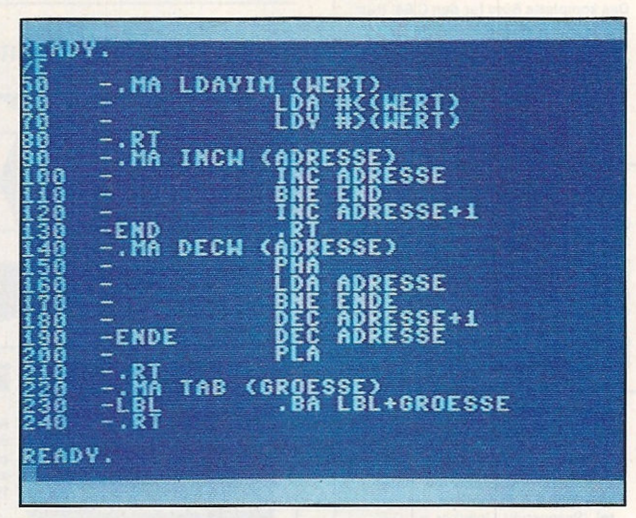
Spiele-Test

So wird bewertet 48
Trends und Flops 48
Master Of The Lamps 48
World Series Baseball 49
Rally Speedway 49
Super Huey 49
Slapshot 50
Crystal Castles 50
Hexenküche 50

Wettbewerbe

Listing des Monats:
Super-Assembler 51
Anwendung des Monats:
Haushaltbuch 52
Terminalprogramm zum Abtippen 149

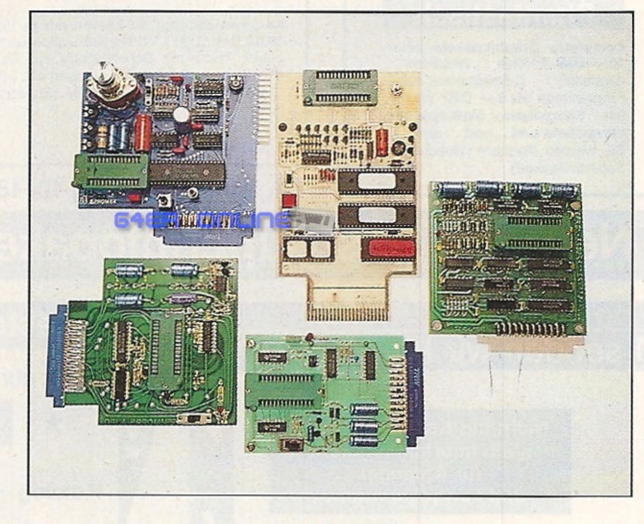
Seite 51



Hypra-Ass — ein Assembler der Spitzenklasse

Hypra-Ass zum Abtippen gehört, was Leistung und Schnelligkeit anbelangt, zu den besten Assemblern, die es zur Zeit für den C 64 gibt. Er assembliert 5 KByte Quelltext in nur vier Sekunden. Die Definition von Makros ist selbstverständlich. Ein tolles Listing des Monats auf Seite 51

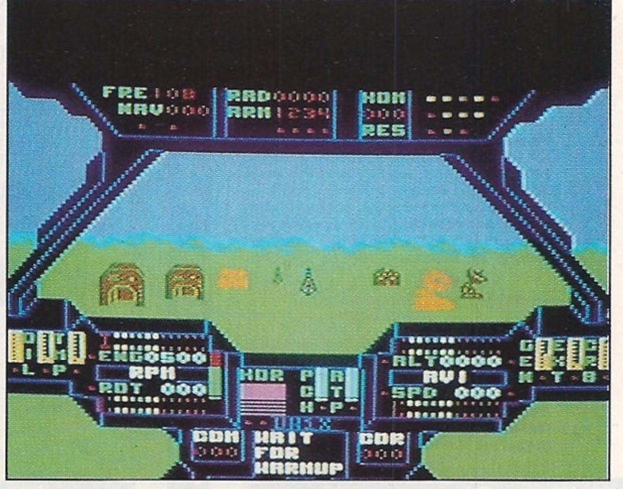
Seite 36



EPROMs — Grundlagen und Programmiergeräte

Was ist eigentlich ein EPROM, und wie programmiert man EPROMs? Dazu brauchen Sie auf jeden Fall ein Programmiergerät. Fünf Vertreter dieser »Prommer« haben wir für Sie getestet. Mechanische Schalter auf diesen Platinen gehören auf jeden Fall der Vergangenheit an.
Seite 36

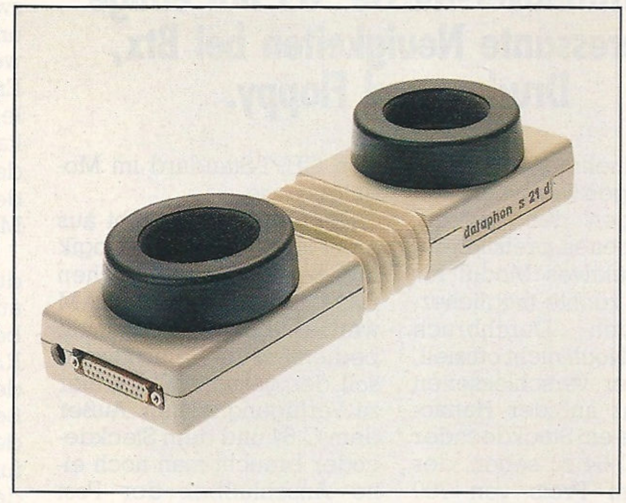
Seite 48



Spiele-Tests

Wir haben uns eine neue Form der Rubrik Spiele-Test ausgedacht. Auf engstem Raum werden die wichtigsten Informationen in übersichtlicher Form dargestellt und zwar für alle Spiele identisch. Damit lassen sich ähnliche Spiele wesentlich besser miteinander vergleichen.
Seite 48

Seite 32



Marktübersicht: Akustikkoppler und Modems

Das Angebot an Akustikkopplern und Modems nimmt mit wachsender Beliebtheit der Datenfernübertragung bei Heimcomputer-Anwendern naturgemäß stark zu. Eine ausführliche Tabelle soll helfen, sich für das richtige Datenübertragungsgerät zu entscheiden.

Seite 32

Seite 118

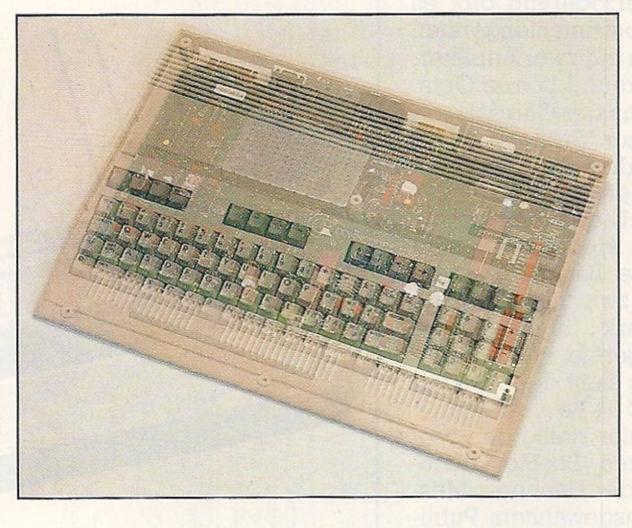


Vergleichstest: Sieben Dateiverwaltungen auf einen Blick

Für die Verwaltung von großen Datenmengen ist der Heimcomputer ideal geeignet, vorausgesetzt man hat das richtige Werkzeug sprich Programm dafür. Wir stellen die sieben bekanntesten Dateiverwaltungsprogramme mit ihren wesentlichen Merkmalen vor.

Seite 118

Seite 17



Erster ausführlicher Testbericht: C 128 (Teil 2)

Im zweiten Teil unseres Tests geht es hauptsächlich um das Betriebssystem CP/M. Außerdem wird gezeigt, welche »Innereien« sich im C 128 befinden und welche Aufgaben die einzelnen Chips haben. Elegant und leistungsfähig präsentierte sich auch das neue Diskettenlaufwerk 1571.

Seite 17

Listings zum Abtippen

Hinweise zum Abtippen 77

Anwendung des Monats
Damit Sie wissen, was Sie ausgeben 54

Listing des Monats
Hypra-Ass — Ein Assembler der Spitzenklasse 66

Tips & Tricks
Basic-Start-Generator 74

So schrumpfen Programme
REM-Killer 75

Komfortable Ein-/Ausgaberoutine 77

Centronics-Interface für jeden Bedarf 78

File-Compactor 82

Elektronischer Merkzettel 83

»Fenster«-Befehle für den C 16 84

Tips & Tricks 85

Terminalprogramm 149

Software-Test

Vergleichstest Datenbanken
Sieben Dateiverwaltungen auf einen Blick 118

Promal im Test
Die ideale Sprache für Profis? 124

Forth-wärts mit M & T-Forth 64 126

Kurse

Assembler ist keine Alchimie (Teil 10) 127

Dem Klang auf der Spur (Teil 7) 132

Memory Map mit Wandervorschlägen (Teil 8) 140

Logeleien (Teil 1) 143

So machen's andere

Commodore-Sportservice — Heimcomputer zur Turnierauswertung 157

Rubriken

Editorial 8

Leserforum 14

Hier gibt's Clubs 76

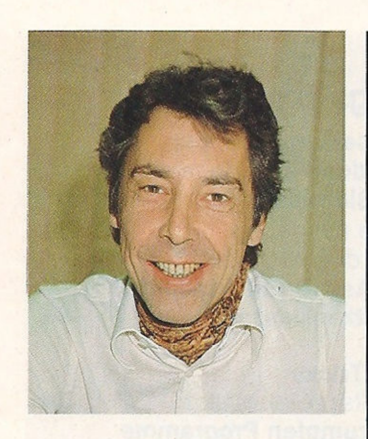
Bücher 88

Leserservice 155

Impressum 163

Vorschau 164





Ein DFÜ-Boom

Die Datenübertragung kommt in Schwung: Unsere Marktübersicht zeigt, wie reichhaltig das Angebot an Akustikkopplern innerhalb eines Jahres geworden ist. Ab etwa 300 Mark gibt es jetzt Geräte mit FTZ-Nummer, beziehungsweise offizieller Zulassung durch die Bundespost — also Geräte, die nicht nur erschwinglich sind, sondern die man auch guten Gewissens benutzen kann.

Parallel dazu gibt es erfreulicherweise auch ein ausreichendes Angebot an brauchbarer Datenübertragungssoftware. Unser Wettbewerb — das Ergebnis finden Sie in dieser Ausgabe — brachte ein sehr komfortables Terminalprogramm, das auch das Laden von Programmen aus einer Mailbox ermöglicht. An diesem Beispiel ist aber auch zu sehen, wie schnell die Entwicklung voranschreitet: Vor drei oder vier Monaten hätte dieses in Basic geschriebene Programm noch voll mit der kommerziell angebotenen Software konkurrieren können. Heute sind die um etwa 150 Mark käuflichen, in Maschinensprache geschriebenen Standardprogramme (oft inklusive Interface) schneller, sicherer und mit mehr Menüpunkten ausgestattet.

Obwohl man sich also ein gutes Terminalprogramm einfach abtippen kann, lohnt es sich immer noch, als Alternative einen Kauf in die Überlegungen einzubeziehen. Auf beiden Gebieten haben Privatleute — bei den Akustikkopplern die Bastler, bei der DFÜ-Software die Programmierer — mit ihren Produkten die kommerziellen Anbieter immer wieder so unter Druck gesetzt, daß zügig sowohl die Preise gesunken als auch die Leistungen gestiegen sind. Es ist zu hoffen, daß dieser produktive Wettbewerb nicht unnötig durch die Postvorschriften gebremst wird.

Michael Pauly, Chefredakteur

Hannover-Messe

Die Hannover-Messe '85 bot einige interessante Neuigkeiten bei Btx, Drucker und Floppy.

Die Teilnehmerzahlen bei Btx liegen weit hinter den Erwartungen der Post zurück. Ein neues, preislich äußerst attraktives Modul für den C 64 könnte möglicherweise den Durchbruch schaffen. Noch nicht offiziell, aber hinter verschlossenen Türen war auf der Hannover-Messe ein Steckdecoder für den C 64 zu sehen, der bei einem Preis von 700 Mark den teueren Btx-fähigen Fernseher und den Decoder ersetzt.

Das Dilemma der verzögerten Einführung von preiswerten Btx-Konsumentengeräten liegt in einem einzigen integriertem Baustein, dem Eurom-Chip. Fast alle Hersteller haben sich bei der Konzeption ihrer Btx-Anlage auf diesen von Valvo zu entwicklenden Chip konzentriert. Bei der jüngsten Eurom-Generation konnten zwar die Darstellungs-Fehler früherer Serien behoben werden. Gleichzeitig aber ist ein neues Problem aufgetaucht: Die aktuelle Serie ist außerordentlich wärmeempfindlich. Wenn es diesen Chips — was sehr schnell geschehen soll — zu warm wird, kann von verläßlicher Funktion nicht mehr die Rede sein.

Durch diese anhaltende Eurom-Verzögerung gibt es bei Konsumentengeräten weiterhin nur zwei Anbieter: Blaupunkt und Loewe Opta. Auch wenn der Chipsatz wie geplant zur Verfügung stünde, die erhofften Preisvorteile würde er einstweilen nicht bringen. Die Preiserwartungen der Hersteller liegen durch die Bank bei einem Aufpreis zwischen 1200 und 1400 Mark für Btx-fähige Fernsehgeräte.

Anders sieht der Sachverhalt für C 64-Besitzer aus.

Technofor stellte den Prototyp ihres Btx-Steckdecoders auf der Hannover-Messe vor ausgewähltem Publikum vor. Es handelt sich dabei um einen vollständigen Decoder nach dem neuesten CEPT-Standard im Modulgehäuse.

Der Decoder besteht aus einer komplexen Videologik und 64 KByte dynamischen RAM. Register und RAM werden über die I/O-Kanäle bedient. Für Telesoftware soll der gesamte Computer zu Verfügung stehen. Außer dem C 64 und dem Steckdecoder braucht man noch eine Anschlußbox der Post (monatliche Gebühren von 8 Mark) oder einen Btx-fähigen Akustikkoppler.

Die abgerufenen Seiten können auf einem handelsüblichen Farbfernsehgerät oder einem Farbmonitor (beispielsweise dem 1701/1702) dargestellt werden. In Zusammenhang mit dem Decoder können alle Schnittstellen und Peripheriegeräte des C 64 benutzt werden. Es ist zum Beispiel möglich, sich Seiten auf Diskette zu überspielen und später wieder anzuschauen oder Batch-Aufrufe (zum Beispiel Wahlprozeduren etc.) wiederholt von der Diskette zu laden und zu starten.

Das Decodermodul ist Telesoftware-fähig (siehe dazu unseren Wettbewerb in der letzten Ausgabe). Software kann so von kommerziellen Anbietern zur ein- oder mehrmaligen Benutzung bezogen werden, oder auch zwischen Btx-Teilnehmern untereinander ausgetauscht werden. Im Herbst, wenn die Entwicklung zweier spezieller Chips abgeschlossen ist, kann der Preis für das Decodermodul nach Angaben des Herstellers unter 500 Mark sinken.

Zusätzlich soll dieses Modul die Auflösung des C 64 auf 480 (600)x240 Punkte verbessern. Zudem kann es als RAM-Floppy eingesetzt werden. Gerade durch die große Verbreitung des C 64 und dem günstigen Anschaffungspreis des Steckdecoders könnte hier die notwendige Initialzündung für eine weite Verbreitung von Btx gegeben werden.

Drucker

Auch bei den Druckern gab es auf der Messe einiges Neues zu bestaunen. Es waren sogar Trends zu erkennen. Die Plotter gehen eine Symbiose mit Schreibmaschinen ein (Bild 1). Panasonic und Silver Reed stellten jeweils entsprechende Modelle vor. Mit vier hochwertigen Kugelschreiberminen schreiben die Printer-Plotter in vier Farben Texte und Grafiken, die auch miteinander mischbar sind.

Silver Reed stellte beispielsweise den Printer-Plotter EB50 vor. Der EB50 besitzt eine Centronics-Schnitt-

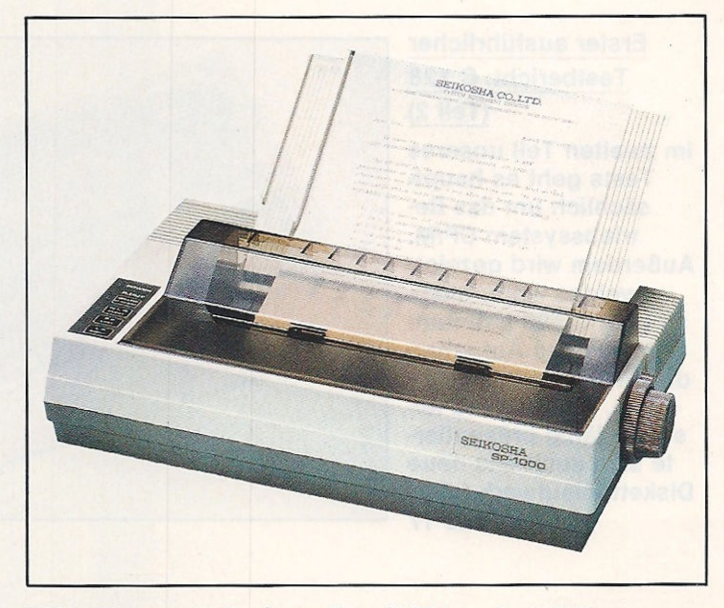


Bild 2. Ein neuer Seikosha, voll grafikfähig und mit NLQ



Bild 1. Eine interessante Synthese: Schreibmaschinentastaturen mit Plotterfunktionen, hier Geräte von Silver Reed und Panasonic

stelle und ist somit mit einem Interface an den C 64 anschließbar. Auf einem 16-Zeichen-Display kann die Eingabe vor dem Druck kontrolliert werden. Schriftarten sind Courier und Italic, in drei verschiedenen Größen mit automatischer Unterstreichung. Auch Balken-, Kurven- und Kuchendiagramme können (auch hier in drei Größen) mit einfachem Tastendruck gezeichnet werden. Der Speicher für diese Grafik hat Platz für 12 Zahlen, also zum Beispiel ausreichend für eine Jahresbilanz. Verarbeitet wird normales DIN-A4-Papier. Zu haben ist der EB50 für ungefähr 800 Mark.

Der Panasonic Printer-Plotter RK-P400C unterscheidet sich vom EB50 unter anderem durch eine 24stellige LCD-Anzeige, 10 verschiedenen Zeichengrößen, einer RS232C-Schnittstelle (ein Kabel zum Anschluß an den C 64 ist auch erhältlich) und einem Preis von 998 Mark inklusive Mehrwertsteuer.

Microscan stellte den neuen Matrixdrucker Seikosha SP-1000VC vor (Bild 2). Seine Daten: direkt anschließbar an den C 64 und die anderen Commodore-Computer (ein Kabel wird mitgeliefert). Wie schon bekannt von Seikosha, besteht eine volle Kompatibilität zum C 64, das heißt, sämtliche Steuerzeichen einschließlich Blockgrafik sowie eigendefinierte Grafik bedeuten keine Probleme beim SP-1000VC. Er beherrscht 10 verschiedene Schriftarten, mit 100 Zeichen pro Sekunde im Normaldruck und bis zu 20 Zeichen pro Sekunde bei Korrespondenzqualität (NLQ = Near Letter Quality). Eine Grafikauflösung bis 1920 Punkte pro Zeile, Druckweg-Optimierung und bidirektionalem Druck sowie Traktor- und Friktionsantrieb sind hervorzuheben. Wichtige Funktionen wie Selbsttest, Zeilen- und Seitenvorschub, linke und rechte Randbegrenzung, Korrespondenz- und Normaldruck, automatischer Einzelblatteinzug sind über die Bediener-Tastatur leicht anzuwählen. Sein Preis: 899 Mark inklusive Mehrwertsteuer.

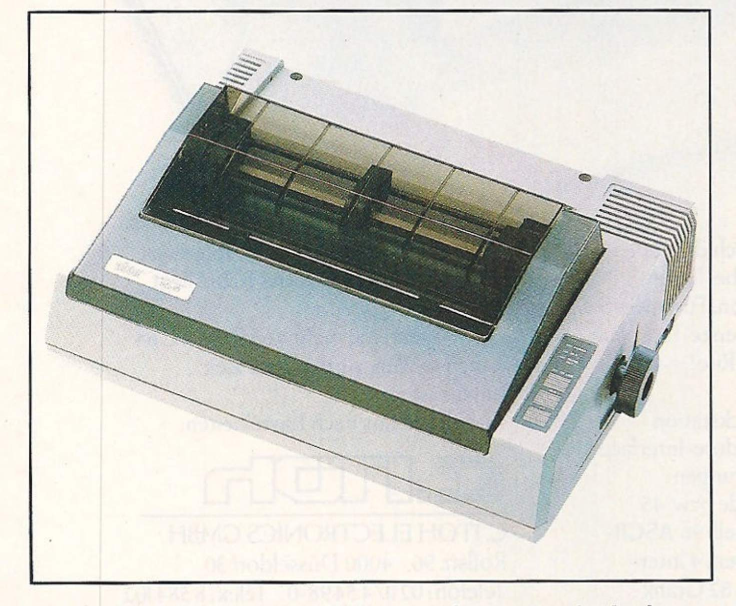


Bild 3. NLQ (Near Letter Quality) setzt sich immer mehr durch, hier der Star SG-10C

Es ist bezeichnend, daß immer mehr Druckerhersteller mittlerweile Drucker mit direktem Anschluß an den C 64 anbieten. So zum Beispiel Star, Epson, Quen-Data, Brother und C.Itoh.

Star präsentierte aus seiner neuen Druckerpalette den SG-10C mit Commodore-Anschluß (Bild 3): Geschwindigkeit: 120 Zeichen pro Sekunde, bidirektional und druckwegoptimiert. Zeichensätze: Schönschrift (NLQ), je 192 Normal- und Grafikzeichen, je 90 Italic- und NLQ-Zeichen und ein frei definierbares Zeichen. Der SG-10C ist voll grafikfähig, mit einer Auflösung von 7x60 Punkte pro Zoll. Doppelt- und dreifachbreiter Druck ist ebenso möglich wie Schmalschrift und reverse Schrift. Die Papierführung ist für Einzelblatt und Endlospapier ausgelegt. Der Preis des SG-10C: Etwas mehr als 1000 Mark.

Mit dem neuen Matrixdrucker GX-80 liefert Epson einen leistungsfähigen und mit unter 1000 Mark auch preiswerten Drucker (Bild 4). Der GX-80 benötigt kein spezielles Interface mehr, sondern ist mit Hilfe von ROM-Modulen bereits fertig an den betreffenden Computer, unter anderem an den C 64, angepaßt. Seine Daten: 100 Zeichen pro Sekunde, 1 KByte Puffer und ein Bereich für einen ladbaren Zeichensatz, und neben dem normalen Schriftbild ein NLQ-Zeichensatz im ROM. Mit 32 internationalen Sonderzeichen deckt er die länderspezifischen Schriftarten ab, und eine Vielzahl programmierbarer Funktionen läßt eine vielseitige Verwendung zu. Der GX-80 transportiert Einzelblätter mittels Friktionsantrieb. Eine Kassette für automatischen Einzelblatteinzug und eine Traktorführung für Endlospapier sind — sehr ungewöhnlich — optional erhältlich.

Der Riteman C+ sieht ganz anders aus, als man es bisher von Matrixdruckern gewohnt ist. Das Papier (Endlospapier oder Einzelblatt) wird nicht von hinten, sondern von vorne in die verstellbaren Traktoren gelegt und läßt sich hinter dem Druckkopf abtrennen. Da der Drucker auf zwei hohen Stelzen stehen kann, bietet sich unter dem Riteman C+ ein idealer Platz an für das Druckerpapier. Das spart eine Menge Platz bei Endlospapier und ist auch für Etiketten- und Einzelblattverarbeitung geeignet. Der Drukker ist MPS 801/803-kompatibel. Die wichtigsten Daten: 105 Zeichen pro Sekunde, 9x9 Matrix und grafikfähig. Schriftarten: Pica, komprimiert, Proportional, Breit-


Der kleine Riese mit
dem schlanken Preis

schriftkombinationen, Super-/Subscript, Unterstreichungen und Unterlängen. Druckarten: Standard, Fettdruck und Doppeldruck. Er besitzt vier Internationale Zeichensätze und 82 Grafiksymbole. Angeschlossen wird der Riteman C+ wie die Commodore-Drucker über ein mitgeliefertes Kabel. Sein Preis: 998 Mark inklusive Mehrwertsteuer.

Quen-Data war mit einem Commodore-kompatiblen Matrix-Drucker, dem DMP 1100 VC vertreten. Die wichtigsten Daten: 100 Zeichen pro Sekunde, deutscher Zeichensatz, Unterlängen, hochauflösende Grafik und Blockgrafik (640x8 Punkte). Die Traktorführung ist verstell- und abnehmbar, aber auch Friktionsantrieb (für Einzelblatt) ist vorhanden. Preis: 849 Mark mit VC-Interface und 1048 Mark mit VC und Centronics-Interface.

Mit der TC-600 hat Brother eine elektronische Schreibmaschine auf den Markt gebracht, die von Brother als Handheld-Telekommunikations-Terminal bezeichnet wird. Und in der Tat ermöglicht die TC-600 den weltweiten Dialog mit allen Computern, die über Telefon erreichbar sind. Texte können aus dem 14,3 KByte Speicher mit 300 Baud gesendet und empfangen werden. Sogar ein eigenes 3½-Zoll-Diskettenlaufwerk ist erhältlich. Und eine Steckdose brauchen Sie auch nicht unbedingt: Batteriebetrieb ist möglich. Der Preis beträgt 1365 Mark plus 169 Mark für das notwendige Kabel. Einen ausführlichen Test lesen Sie bald im 64'er Magazin.

Gleich 3 Drucker von Citizen waren auf dem Synelec-Stand zu sehen. Der MSP-10 (1699 Mark) und sein breiterer Bruder, der MSP-15 (2199 Mark) präsentierten sich im wirklich schönen Slim-Line-Design. 160 Zeichen pro Sekunde, NLQ-Schrift und viele Zeichensätze plazieren die beiden in die gehobene Leistungsgruppe. Im Low-cost-Bereich bewegt sich der Citizen TTP 680 mit 600 Mark. Der Thermodrucker erreicht eine Druckgeschwindigkeit von 50 Zeichen pro Sekunde und kann direkt an den C 64 angeschlossen werden.

Auch in bezug auf Farbe tut sich was auf dem Drucker-Markt. Der Seikosha GP-700 hat ein VC hinter seinen Namen bekommen, will heißen: direkt anschließbar an den C 64. Mit den vier Grundfarben lassen sich insgesamt sieben Farben darstellen. Preis: 1198 Mark inklusive Mehrwertsteuer.

Mit ungefähr 2300 Mark bewegt sich der Farbdrukker 5510/5520 von Juki zwar in einer anderen Preisklasse, leistet jedoch um einiges mehr. Mit 180 Zeichen pro Sekunde wirklich schnell und mit weniger als 60 db(A) noch angenehm leise, einem 2 KByte Puffer und eine ganzen Menge an Schriftarten und Zeichensätzen.

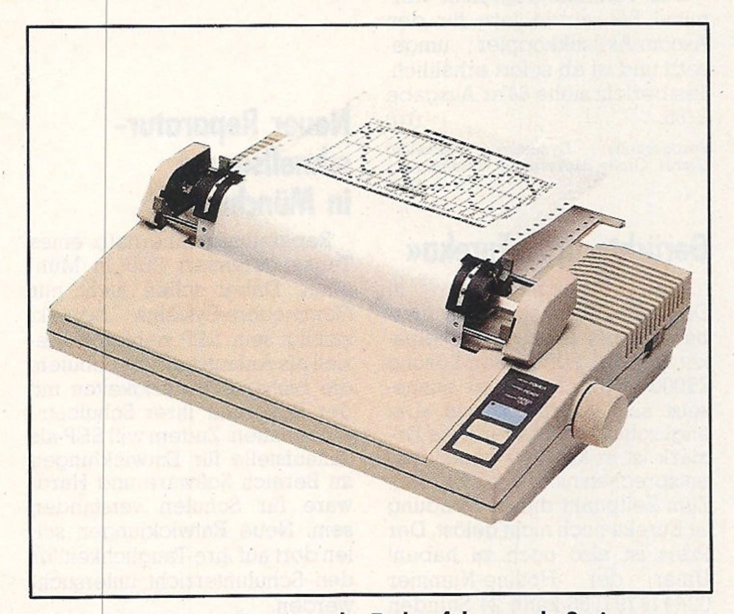


Bild 4. Beim Epson GX-80 ist die Traktorführung als Option erhältlich

Diskettentrends

Die Hannover-Messe zeigte es ganz deutlich: Auf dem Diskettensektor tut sich einiges. Neue Technologien und neue Formate sind im Vormarsch. Dabei werden die Disketten immer kleiner und die Speicherkapazität steigt.

Alle wichtigen Diskettenhersteller bieten inzwischen 3½-Zoll-Disketten an. Die Entscheidung von IBM (beim AT) für 3½-Zoll wird wahrscheinlich den letzten Anstoß zur allgemeinen Einführung dieses Formates geben (die anderen Formate im 3-Zoll-Bereich wie 3, 3¼ oder 3¾ dürften sich wohl nicht durchsetzen). Die 3½-Zoll-Diskette hat tatsächlich einige Vorteile gegenüber dem 5¼-Zoll-Format: Sie ist handlicher, paßt in jede Hemdtasche und ist durch das stabile Gehäuse mit der verschließbaren Schreib-/Leseöffnung optimal gegen Verschmutzung und Beschädigung gesichert. Die Speicherkapazität steht dabei den 5¼-Zoll-Disketten in nichts nach, da durch den besseren Schutz der Magnetschicht höhere Aufzeichnungsdichten möglich sind. Man darf gespannt sein, wann das erste 3½-Zoll-Laufwerk für den C 64 erhältlich sein wird.

Im Hinblick auf neue Technologien scheint sich für die Diskette der gleiche Weg wie bei der Compact Cassette im HiFi-Sektor abzuzeichnen: Bei BASF setzt man auf Chromdioxid- und Reineisen-Disketten zur Steigerung der Aufzeichnungsdichte und -qualität. Maxell plant ebenfalls die Einführung einer Reineisendiskette mit einer Kapazität von fünf Megabyte pro Diskettenseite. Auch von anderen Herstellern wird dieser Schritt in Erwägung gezogen.

Auch optische Disketten, bei denen die Information wie bei einer Videoplatte per Laserstrahl abgetastet wird, sind für die Zukunft möglicherweise auch im gehobenen Homecomputer-Bereich zu erwarten. Speicherkapazitäten um die 2,6 Gigabyte, also etwa »10600 Blocks free« — das läßt wohl des Herz jedes C 64-Besitzers höher schlagen.

(aa/gk/ev)

James Bond — Computerspiel und Film

Eine große Fabrik in Sunvalley, die Computerchips herstellt, soll in die Luft gesprengt werden. Agent 007 (Roger More) soll dies verhindern. Soweit die Rahmenhandlung des neuen James Bond-Filmes »A View to kill« (Im Angesicht des Todes). Der Film wird am 12. Juni in England anlaufen. Eine Woche vorher, am 5. Juni, erscheint ein Computerspiel der englischen Firma Domark zu diesem Film.

Das Spiel wird aus drei Teilen bestehen, die genau an die Filmhandlung angelegt sind. Der erste Teil spielt in Paris. Im Restaurant des Eifelturms wird in James Bonds Gegenwart ein Informand erschossen. Der Mörder flüchtet dann in einem Auto quer durch Paris. Hier setzt der erste Spielteil ein. Sie müssen als 007 den Wagen verfolgen. Auf dem Monitor sehen Sie den Stadtplan von Paris, auf dem die Position Ihres und des anderen Autos gekennzeichnet ist. Weiterhin wird in 3D-Grafik der Ausblick aus dem Wagen gezeigt. Die Verfolgungsjagd kann beginnen. Schaffen Sie diesen Teil nicht gleich auf Anhieb, langweilig wird das nächste Spiel bestimmt nicht. Der Fluchtweg des Autos verändert sich mit jedem Spiel.

Haben Sie diese Hürde genommen, erhalten Sie ein Paßwort, das sich aus dem vorher eingegebenen Namen und der benötigten Zeit ergibt. Mit diesem Code gelangen Sie in den nächsten Teil des Spiels.

Mit diesem Teil gelangen Sie nach San Francisco. Sie befinden sich in einem brennenden Hochhaus. Dieser Spielteil ist ein Arcade-Adventure. Sie müssen in fünf Stockwerken mit insgesamt 75 Räumen Hilfsmittel suchen, um aus dem brennenden Haus zu entkommen. Doch achten Sie darauf: das Feuer frißt sich schnell weiter vor.

Auch hier bekommen Sie nach erfolgreicher Beendigung des Levels ein Paßwort für den dritten Teil des Spieles.

Im dritten Abschnitt befinden Sie sich in einem Höhlensystem unter dem Sunvalley. Sie müssen eine Bombe finden, entschärfen und sie aus dem Höhlensystem entfernen. Ähnlich wie bei Impossible Mission müssen Sie vorher in verschiedenen Räumen Aufgaben lösen. Dieser Teil ist unter anderem ein Wettlauf gegen die Zeit. Entschärfen Sie die Bombe nicht rechtzeitig, ist Sunvalley verloren.

Am Ende des Spieles, so ist von Domark zu hören, befindet sich eine »secret animated sequence«. Genaueres erfahren Sie in der nächsten Ausgabe.

Musik von Duran Duran

Die Titelmusik zum neuen James Bond-Film stammt von Duran Duran. Es gelang dem englischen Software-Produzenten, die Rechte an der Musik für ihr Spiel zu bekommen. Ob die Umsetzung auf den C 64 genauso gelungen ist wie bei Ghostbusters, bleibt abzuwarten. Auch dies erfahren Sie in der nächsten Ausgabe.

Bei dem Spiel handelt es sich übrigens tatsächlich um eine Neuentwicklung, die sich auf den neuen Film bezieht. Sie sollte also nicht mit dem Spiel 007 von Parker Brothers verwechselt werden, das wohl als sehr verunglückt bezeichnet werden konnte und keinen Bezug auf einen James Bond-Film hatte.

Der Trend

Stars auf dem Bildschirm zu haben, war schon immer das Anliegen der Filmproduzenten. Doch auch die Softwareproduzenten wollen sich ein Stück des Erfolgskuchens abschneiden. Spätestens nach dem legendären Film-, Musik- und (natürlich) Softwareerfolg von »Ghostbusters« wollen alle auf der Erfolgswelle mitreiten. Nach dem Beginn der »Spiel zum Buch«-Welle kommt jetzt also auch die »Spiel zum Film«-Welle ins Rollen. Lizenzen werden nicht nur zu den Markennamen der Stars und zur Filmhandlung gekauft, sondern auch der Musik. Erst Ghostbusters, jetzt Duran Durans James Bond-Musik. Doch bei allem Trend: Warten wir lieber immer erst ab, ob das Spiel soviel Spaß bringt wie das Ansehen des Films. Der Name eines Stars macht noch lange kein gutes Spiel aus. (rg)

Noch ein Floppy-Beschleuniger

Das System »Floppy-Flash 1541« von Mikrotronic soll das Laden und Speichern aller File-Typen auf dem Laufwerk 1541 nach Angaben des Hersteller bis zu 10mal schneller machen. Das System wird am Expansion-Port angeschlossen (der durchgeschleift ist) und bietet erweiterte DOS-Befehle auf Funktionstasten, einen speziellen Reset und ein Centronics-Interface mit Hardcopy-Funktion. Die RS232-Schnittstelle soll verbessert worden sein. Außerdem sollen ein Schnellkopierer und eine Schreibschutzfunktion eingebaut sein.

Die Umrüstarbeiten sollen weniger als 5 Minuten in Anspruch nehmen. Zum Preis von 79 Mark ist eine Erweiterung erhältlich, die den Betrieb eines zweiten Laufwerks gestatten soll. Außerdem ist ein Ein-Minuten-Schnellkopierer für 39 Mark lieferbar.

Info: Dipl.-Ing. K. Roreger, Liebigstr. 28, 4780 Lippstadt, Tel. 02238/43556

VAM ruft alle mit Computern unterversorgten Schulen zur Mitarbeit auf

Viele Schulen im Bundesgebiet verfügen noch nicht über eine ausreichende Ausstattung an Computern, um ihren Schülern eine zeitgemäße Ausbildung an diesen Systemen zu gewährleisten. In einigen Bundesländern sind zwar bereits Aktionen auch von seiten der Kultusministerien in Angriff genommen worden, um dieser Unterversorgung abzuhelfen. Dennoch verbleibt die Initiative, die jeweilige Schule mit einer ausreichenden Anzahl an Computern zu versorgen, immer noch den Lehrern, Direktoren, engagierten Schülern oder Eltern vorbehalten.

Eines hat sich auf dem Technologie-Forum, das im März '85 von Commodore veranstaltet wurde, ganz deutlich gezeigt: Der Computer wird sicher die Kreidepädagogik ersetzen, doch die Rolle des Lehrers wird in absehbarer Zeit nicht in Frage gestellt sein. Deshalb sind gerade die Lehrer, welche einen »Notstand« an ihrer Schule erkannt haben, aber auch die, die bisher noch mit einer gewissen Skepsis und Reseviertheit dem »neuen Medium« Computer gegenüberstanden, aufgerufen, den Schülern (und sich selbst) zeitgerechte Lehr- und Lernmittel zur Verfügung zu stellen.

Die Idee des VAM (Verein zur beratenden Aufklärung über die Mikrocomputertechnologie e.V.), möglichst in alle Schulen eine ausreichende Anzahl von Computern zu bringen, beruht auf der sogenannten Mietpatenschaft.

Dabei werden acht bis zehn spezielle Schulcomputer »Leibniz I« (momentan basierend auf dem C 64, der in Zukunft aber um einige Leistungsmerkmale wie Vernetzung oder Schnelladesysteme bereichert werden wird) zur Verfügung gestellt. Handwerksbetriebe, Banken, Versicherungen, mittelständische Unternehmen oder Kaufhausketten in der Umgebung der Schule übernehmen die (monatliche) Finanzierung eines oder mehrerer dieser Systeme.

Die Aktivierung dieser Unternehmen wird von den jeweiligen Initiatoren und dem VAM vorgenommen. Alle Beteiligten haben von dieser Mietpatenschaft Vorteile: Die Schulen kommen zu ihren Computern und können den Schülern zeitgemäßen Unterricht präsentieren (nicht nur in ureigensten Gebieten wie Informatik oder Mathematik, sondern auch als unterstützende Maßnahmen für eine moderne Aufbereitung der Lehrinhalte in anderen Fächern). Die Firmen der Umgebung können mit Schulabgängern rechnen, die hervorragende Kenntnisse in den neuen Formen der Datenver- und bearbeitung und der Computertechnologie aufwarten können. Für eine erste Kontaktaufnahme steht der VAM bereit.

Info: VAM, Rheinallee 6, 5300 Bonn 2, Tel. 0228/363571

Neues Kopierprogramm

Die »Orginal Data Backup Box« ist ein 40-Spur-Error-Kopierer von Mikrotronic für 139 Mark auf Modul, das 40 Spuren und Read Errors bei unterschiedlicher Synchronisation kopieren kann. Dieses Modul enthält DOS-Befehle, Directory-Anzeige, 40-Track-Error-Backup in drei Minuten, ein Zwei-Minuten-Fast-Backup für 36 Spuren und schnelles Filecopy für ungeschütze Files. Nach Angaben des Vertreibers richtet sich dieses Modul an alle Besitzer teurer Original-Software, die Ihre Disketten im Dauereinsatz haben. Als »Original Read-Error Box« wird für 89 Mark eine Version angeboten, die sich nicht auf die Synchronisation der Spuren 37 bis 40 bezieht.

Info: Dipl.-Ing. K. Roreger, Liebigstr. 28, 4780 Lippstadt, Tel. 02238/43556

Software für Friseure

Eine spezielle Brachenlösung für Friseure bietet der H.G.M. Alpenland-Vertrieb an. Das gesamte System, bestehend aus C 64, Datenspeicher, Bildschirm, Drucker, Kassendrucker und diversen Programmen wie Registrierkassen, Auswertung mit Leistungsstatistik für einen Salon und 50 Mitarbeiter, Lohn/Grafik, Hauptprogramm mit ständiger Inventur für Verkauf und Kabinett, Initialisierung, Summierung und Kassenzettel kostet 4990 Mark. Die gesamte Anlage kann auch für eine monatliche Gebühr von 120 Mark gemietet werden.

Info: H.G.M. Alpenland-Vertrieb, G. Linder, Hauptstraße 43, 8222 Ruhpolding

Einbaumodem auch für den C 64

In Kürze soll es laut Bundespost auch für den C 64 das Einbaumodem geben. Der Termin der Einführung stand zum Redaktionsschluß noch nicht fest, wohl aber der Preis. Das Einbaumodem kostet pro Monat 20 Mark Miete. Der Anschluß erfolgt durch die Bundespost. Genauere Angaben in einer der nächsten Ausgaben. (rg)

Terminal 64 für Ascom-Akustikkoppler

Das Terminalprogramm »Terminal 64« wurde jetzt für den Ascom-Akustikkoppler umgesetzt und ist ab sofort erhältlich. Testbericht siehe 64'er Ausgabe 2/85. (rg)

Bezugsquelle: Dynamics Marketing GmbH, Große Bäckerstr. 11, 2 Hamburg 1

Gerüchte um »Eureka«

Die Gerüchteküche in Deutschland kocht. Es geht hierbei um das Abenteuerspielpaket Eureka, auf dessen Lösung 25000 englische Pfund ausgesetzt sind. Gerüchte wie »Die englische Veranstalterfirma Domark ist in Konkurs gegangen« entsprechen nicht der Wahrheit. Zum Zeitpunkt dieser Meldung ist Eureka noch nicht gelöst. Der Preis ist also noch zu haben! Unter der Hotline-Nummer 0044 1879 11 66 kann 24 Stunden lang der aktuelle Stand abgerufen werden. (rg)

53-KByte-Terminalprogramm

Das Terminalprogramm »Terminal 86« gibt es auf EPROM und stellt 53 KByte für die Datenaufnahme zur Verfügung. Eine ständige Helpseite ist integriert und jederzeit aufrufbar. Die Übertragungsrate ist wählbar zwischen 300, 600 und 1200 Zeichen pro Sekunde. Ebenso kann man zwischen ASCII oder deutscher Tastatur wählen. Das Programm verfügt über einen eingebauten Editor, die Parametereinstellung (Parität, Voll- oder Halbduplex, etc.) erfolgt über Tastendruck und kostet 139 Mark.

Info: JJC, Crispinstr. 4, 4600 Dortmund 50

Neuer Reparaturschnellservice in München

Reparaturen innerhalb eines Tages garantiert SEP in München. Dabei sollen nicht nur Commodore-Systeme berücksichtigt sein. SEP sieht sich speziell als Anlaufstelle für Schulen, die bisher Schwierigkeiten mit der Reparatur ihrer Schulcomputer hatten. Zudem will SEP als Anlaufstelle für Entwicklungen im Bereich Software und Hardware für Schulen verstanden sein. Neue Entwicklungen sollen dort auf ihre Tauglichkeit für den Schulunterricht untersucht werden.

Info: SEP, Hans-Pinsel-Str. 10b, 8013 Haar bei München, Tel.: 089/4605340





Ärger mit der 1541

Ich habe einige Schwierigkeiten mit meinem Floppy-Laufwerk. Wenn es längere Zeit in Betrieb und ziemlich warm geworden ist, nimmt die Diskette kein Programm mehr an. Beim Speichern meldet die Floppy dann Fehler 20 oder 27. Das Laufwerk lädt dann auch nicht mehr, auch keine Directories von anderen Disketten. Nach längerem Abkühlen kann wieder geladen werden, aber nicht mehr die Programme, die ich vorher abspeichern wollte. Auf dem Laufwerk ist noch Garantie und es war auch schon zur Reparatur beim Händler. Der sagte, es sei eine Neueinstellung vorgenommen worden. Aber der Fehler tritt immer wieder auf. Schadet es der Floppy, wenn ich einen Reset auslöse? Dabei läuft die Floppy kurz an und die rote Lampe leuchtet. *Josef Spiertz*

Ihr Problem beruht auf der thermischen Ausdehnung von Metallen. Die Mechanik des Laufwerkes dehnt sich normalerweise nur minimal aus. Bei Ihnen scheint der unglückliche Extremfall vorzuliegen, daß durch diese Ausdehnung der Schreib-/Lesekopf total verstellt wird. Dann kann die Floppy-Station weder Daten lesen noch schreiben. So, wie bei Ihnen geschildert, darf dieses Problem aber unter normalen Umständen nicht auftreten. Normalerweise ist ein mehrstündiger Betrieb, solange nicht laufend formatiert wird, ohne Probleme möglich. Es gibt hier zwei Lösungsmöglichkeiten: Entweder Sie sorgen für Kühlung, indem Sie die Floppy »offen«, das heißt ohne Gehäusedeckel, betreiben oder einen Lüfter verwenden. Die Alternative wäre, daß Sie auf einen Umtausch des Laufwerkes bei Ihrem Händler bestehen.

Bei einem Reset am Computer wird auch in der Floppy ein Reset ausgelöst. Das hat keinerlei schädliche Folgen für Floppy und Diskette.

Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

Vizawrite-Hilfe

Ich arbeite mit einer englischen Version von Vizawrite und suche nach einer Möglichkeit, deutsche Umlaute darstellen zu können.
Ausgabe 5/85

Bertram Hafner

Der Zeichenspeicher bei Vizawrite 64, englische Version, beginnt bei $3800 und endet bei $3FFF. Durch entsprechende POKEs können in diesem Bereich die notwendigen Änderungen vorgenommen werden. Dies ist bei Ä,Ö,Ü ohne weiteres möglich, indem man die Zeichencodes für die eckigen Klammern beziehungsweise das Pfundzeichen entsprechend ändert.

Leider stößt ein ähnliches Vorgehen bei den kleingeschriebenen Umlauten ä,ö,ü sowie bei ß auf erhebliche Schwierigkeiten, da aufgrund bestimmter AND-Verknüpfungen innerhalb des Programms die Erzeugung der entsprechenden Zeichencodes durch Tastendruck verhindert wird.

Ich selbst habe die notwendigen Änderungen in Vizawrite 64 unter großen Schwierigkeiten vorgenommen, so daß nun auf Bildschirm und Drucker alle Umlaute und ß in Verbindung mit der Commodore-Taste erzeugt werden können.

In Verbindung mit Hypra-Load kann nun sogar die nervtötende Ladezeit um den Faktor sechs verkürzt werden.

Ich bin gerne bereit, Besitzern des englischen Vizawrite ihr Originalprogramm gegen eine Unkostenvergütung entsprechend zu modifizieren.

Hanno Hettinger

Wer kennt das Quick Data Drive?

Ich möchte mir im Sommer dieses Jahres das Quick Data Drive von NCS zulegen. Da diese Floppy-Alternative noch ziemlich wenig verbreitet ist, wäre ich an Erfahrungen anderer Leser mit diesem System interessiert. *Marc Nelles*

Listschutz geknackt

In der Ausgabe 12/84 des 64'er-Magazins wurde ein laut Autor »unknackbarer« Listschutz vorgestellt. Wer sein Programm wirklich wirksam schützen will, der sollte diesen »Schutz« allerdings nicht verwenden. Er kann nämlich durch einen einfachen POKE-Befehl (POKE 43,82) außer Gefecht gesetzt werden. Dieser POKE bewirkt ein Verschieben des Basic-Starts, so daß beim Listen die beiden Zeilen, in denen der Listschutz aktiviert wird, einfach übersprungen werden.

Der eben genannte POKE gilt zwar nur ganz speziell für das abgedruckte Listschutzprogramm, aber selbst ein mittelmäßiger Hacker kann derartige Schutzmaßnahmen leicht knacken, und sei es etwas aufwendiger mit einem Maschinensprache-Monitor.

Stefan Pastuszka

Zum Thema Listschutz erreichen uns ständig neue Zuschriften. Viele Leser suchen einen guten Listschutz für ihre Programme, ohne zu bedenken, daß ein abgedruckter Listschutz kein Schutz mehr ist, denn aus der Kenntnis der Funktionsweise ergibt sich fast zwangsläufig auch die Möglichkeit, einen solchen Schutz auch zu »knacken«, wie das obige Beispiel beweist.

Hi-Eddi mit CP 80

Das wirklich gute Grafikprogramm Hi-Eddi (Listing des Monats in Ausgabe 1/85) kann auch mit einem Melchor CP 80-Drucker zusammenarbeiten. Durch folgende Änderungen im Hi-Print ist eine Druckausgabe inklusive aller Sonderfunktionen auf dem CP-80 mit Vobis-Interface möglich:
460 DATA 4
550 DATA 27,51,21,255,255
580 DATA 27,75,255,255,255
Ärgerlich ist bei Hi-Eddi jedoch, daß man im Sprite-Edit-Modus die gesetzten Punkte nicht wieder löschen kann.

Bei meinem C 64 taucht noch ein spezielles Problem auf: Direkt nach dem Einschalten sind die Grafiken von verschiedenen Spielen total verzerrt und verschoben. Nach einiger Zeit gibt sich das allerdings wieder von selbst. Wer hat ein ähnliches Problem oder kann mir helfen?
Thomas Küpper

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene, dann schreiben Sie uns. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

Software aus Amerika

Ich möchte für meinen C 64 Programme direkt aus Amerika besorgen. Meine Frage: Läuft die in Amerika angebotene Software auch auf der europäischen Version des C 64 ? Ich denke dabei insbesondere an Probleme, die sich aus unterschiedlichen TV-Normen (Pal/NTSC), Netz- und Quarzfrequenzen ergeben können.
Ingo Zeigler

Fast die gesamte in Amerika angebotene Software läuft völlig problemlos auch auf der europäischen Version des C 64. Die unterschiedlichen TV-Normen bedingen lediglich unterschiedlich konzipierte HF-Modulatoren, haben jedoch keinen Einfluß auf die Software. Die unterschiedlichen Netz- und Taktfrequenzen dürften nur bei sehr wenig Programmen wirklich Probleme machen. Es läßt sich aber nicht pauschal sagen, bei welcher Art von Programmen Schwierigkeiten auftreten könnten. Im Einzelfall hilft leider nur Ausprobieren.

Leser fragen — Willi Brechtl antwortet

Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.

Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.

Hardware-Sorgen

1. Kann ein Schaden an den Geräten (Zentraleinheit, Floppy und Farbfernseher) entstehen, wenn ich sie alle drei auf einmal einschalte oder muß ich erst den Fernseher einschalten, warten bis ein Bild da ist, dann die Zentraleinheit in Betrieb setzen und nachher das Floppy-Laufwerk einschalten?

2. Was kann an der Floppy-Station und der Diskette kaputtgehen, wenn sich eine solche während des Ein- oder Ausschaltens im Laufwerk befindet? Mir ist das schon öfter passiert, aber soweit ist noch alles ganz.

3. Kann es sein, daß es den C 64 mit verschiedenen Betriebssystemen gibt? Mein Freund hat ein Spiel »Popeye«, das bei ihm einwandfrei läuft. Wenn wir es aber bei mir spielen wollen, macht es immer einen Aussteiger.

4. Was kann an der Hardware kaputtgehen, wenn ich während des Betriebs ein Verbindungskabel löse?

Michael Reichel

1. Den Geräten ist es völlig egal, in welcher Reihenfolge sie eingeschaltet werden. Besonders bequeme Menschen wie beispielsweise die 64'er-Redakteure lösen das Einschaltproblem besonders elegant: Alle Geräte werden an eine Steckdosenleiste mit Schalter angeschlossen. Liegt diese auch noch auf dem Boden, läßt sich mit einem einfachen Fußdruck die gesamte beschriebene Hardware (auch Drucker) einschalten. Vorsicht ist allerdings bei älteren Fernsehgeräten geboten. Diese vertragen es teilweise nicht, wenn die Netzspannung direkt und nicht über den eingebauten Schalter gesteuert wird.

2. Am Floppy-Disk-Laufwerk wird normalerweise kein Schaden entstehen. Disketten können aber theoretisch durch die plötzliche Spannungsspitze im Gerät teilweise gelöscht werden. Uns ist das bisher auch noch nicht passiert, trotzdem sollte hier Vorsicht geboten sein.

3. Es gibt verschiedene Versionen des 64er. Die Unterschiede sind allerdings minimal und treten nur vereinzelt zu Tage. So gibt es mehrere Ausführungen der CPU, was sich bei einigen sogenannten »illegalen Opcodes«, also eigentlich nicht vorgesehenen Maschinenbefehlen bemerkbar macht. Auch die Bus-, Sound- und Videobausteine werden in verschiedenen, aber fast identischen Versionen gefertigt.

4. Das sollte man tunlichst unterlassen. Gerade die Busbausteine in der 1541-Floppy sind da sehr empfindlich. Besonders gefährlich sind hier statische Aufladungen. Wer beim Umstecken seine Floppy zerstört, weil er in geladenem Zustand die Ports berührt, der darf mit Reparaturkosten bis zu 100 Mark und mehr rechnen. Das gleiche gilt für Computer (Serieller Bus, Datasetten-Anschluß, Userport und sogar Joystickanschluß) und Drucker.

Von Basic zur Maschinensprache?

Ist es möglich, ein Basic-Programm mit Hilfe eines Maschinenprogramms in ein Maschinenprogramm zu verwandeln? Wenn ja, womit?

Frank Wild

Solche Programme gibt es tatsächlich. Sie heißen »Basic-Compiler«. Ein solcher Compiler übersetzt ein Basic-Programm in Maschinensprache, die der Computer direkt verstehen und ausführen kann. Derart übersetzte Programme sind daher um einiges schneller als entsprechende reine Basic-Programme. Einen ausführlichen Test verschiedener Basic-Compiler für den C 64 finden Sie in den 64'er-Ausgaben 2/85 und 4/85.

Was ist der MSE?

Ich verstehe den Aufbau der MSE-Sprache nicht. Warum sind in einer Zeile jeweils neun verschiedene Befehle?

Tim Wensky

»MSE« ist keine eigene Computersprache, sondern nur ein Eingabesystem für Programme in Maschinensprache. Diese werden in einer besonderen Form als »Hexadezimalzahlen« abgedruckt. Dieses Zahlensystem ist nicht wie die uns geläufigeren Dezimalzahlen auf der Basis 10 aufgebaut, sondern auf der Basis 16. Als zusätzliche Ziffern werden dabei einfach die ersten Buchstaben des Alphabets genommen. Statt dezimal ... 8,9,10,11,12,13,14,15,16,... wird hexadezimal folgendermaßen gezählt: ...8,9,A,B,C,D,E,F,10,...

Jeweils zwei Hexadezimalziffern bilden ein »Byte«, also eine Zahl zwischen 0 und 255 (hexadezimal 00 und FF). Ein Byte entspricht also dem Inhalt einer Speicherzelle oder auch einem Maschinenbefehl. In einer MSE-Zeile stehen jeweils acht Programm-Bytes. Die letzte Zahl ist eine Prüfsumme. Sie wird verwendet, um Fehleingaben in einer Zeile weitgehend auszuschließen.

Bitte lesen Sie beim Abtippen von MSE-Programmen die zugehörige Bedienungsanleitung besonders aufmerksam durch. Versuchen Sie insbesondere nie, mit dem MSE eingegebene Programme einfach mit LOAD zu laden und mit RUN zu starten. Vielmehr müssen Sie MSE-Programme (wie alle Maschinenprogramme) mit »LOAD "name",8,1« laden.

63299 Bytes free?

Ich habe den Tip bekommen, daß die Befehlsfolge »POKE 56,255 : SYS 58234« beim VC 20 einen freien Speicherplatz von über 61000 Bytes ergeben soll. Ich habe nun diese Befehlsfolge auf meinem C 64 eingegeben und erhalte tatsächlich nach der Abfrage »PRINT FRE(0) + 2116« die Antwort »63229«.

Sind diese 63229 Bytes wirklich frei zum Programmieren und wird dadurch auch keine andere Funktion beeinflußt?

Torsten Niek

Na, da hat Sie aber wirklich einer an der Nase herumgeführt. Der POKE-Befehl setzt zwar (allerdings nur rein rechnerisch) die Speichergrenze für den Basic-Interpreter nach oben, aber sonst tut sich überhaupt nichts. Sie haben weiterhin ihre knappen 38 KByte Basic-Speicher, weil Sie mit einem POKE-Befehl schwerlich Hardware-Aufbau und Betriebssystem eines Computers ändern können. Im Gegenteil, wenn Sie versuchen mit Variablen zu arbeiten, kann das sogar zu einem Absturz des Computers führen, weil dieser verzweifelt versucht, Variableninhalte im ROM abzulegen. Dies kann natürlich nicht gut gehen.

Der SYS-Befehl hinter dem POKE ist übrigens reine Augenwischerei, denn der bewirkt nur einen Sprung in die NMI-Routine, also dasselbe, was beim Drücken der Tastenkombination RUN/STOP-RESTORE passiert.

Eine Möglichkeit, noch mehr Basic-Speicher zu erhalten, ist die Verwendung von speziellen Erweiterungen wie das Business-Basic-Moduls von Kingsoft. Einen Test hierzu finden Sie in dieser Ausgabe.

Graphics Basic verschenken?

Ich interessiere mich für »Graphics Basic« zum C 64 und habe deshalb folgende Fragen:

1. Kann man jetzt, nachdem die Herstellerfirma HES Konkurs gemacht hat, Kopien von Graphics Basic unter Freunden verteilen oder sogar an Außenstehende verkaufen?

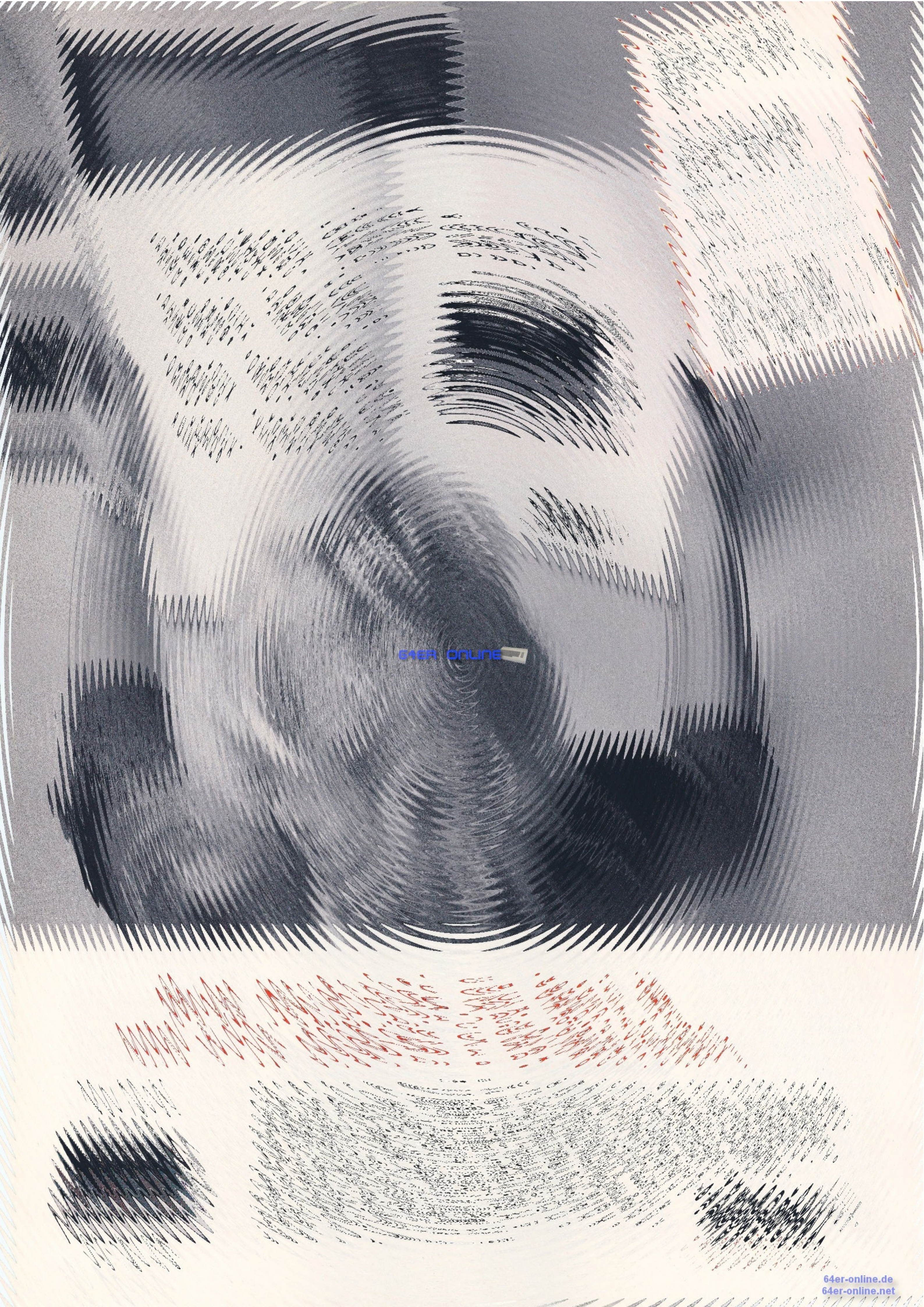
2. Wo kann man das Handbuch zu Graphics Basic zwecks Information bestellen?

Markus Emmert, Sven Schal

1. Gegenfrage: Wenn das Volkswagenwerk Konkurs angemeldet hätte, würden Sie dann nach Wolfsburg fahren, sich ein paar Golfs aus dem Lager holen und die Fahrzeuge unter Freunden verteilen oder an Außenstehende verkaufen?

Also nochmals in aller Deutlichkeit: Sie dürfen fremde Software, egal ob tatsächlich kommerziell vertrieben oder nicht, nicht ohne ausdrückliche Genehmigung des Urhebers kopieren und an andere Personen weitergeben. Wenn Sie das Original haben, dürfen Sie davon zwar Kopien anfertigen, aber ausschließlich für den eigenen, privaten Gebrauch (Sicherheitskopien). Sie dürfen derartige Kopien nicht verkaufen, tauschen oder verschenken.

2. Da Graphics Basic in Deutschland nicht mehr vertrieben wird, ist auch das Handbuch nicht mehr erhältlich.



Erster ausführlicher Test: C 128

Teil 2

In der letzten Ausgabe haben wir Ihnen die Eigenschaften und Möglichkeiten des C 128 vorgestellt. Diesmal sollen Sie erfahren, was alles im C 128 und dem neuen Diskettenlaufwerk 1571 zu finden ist, und wie es um das Betriebssystem CP/M im 128er bestellt ist.

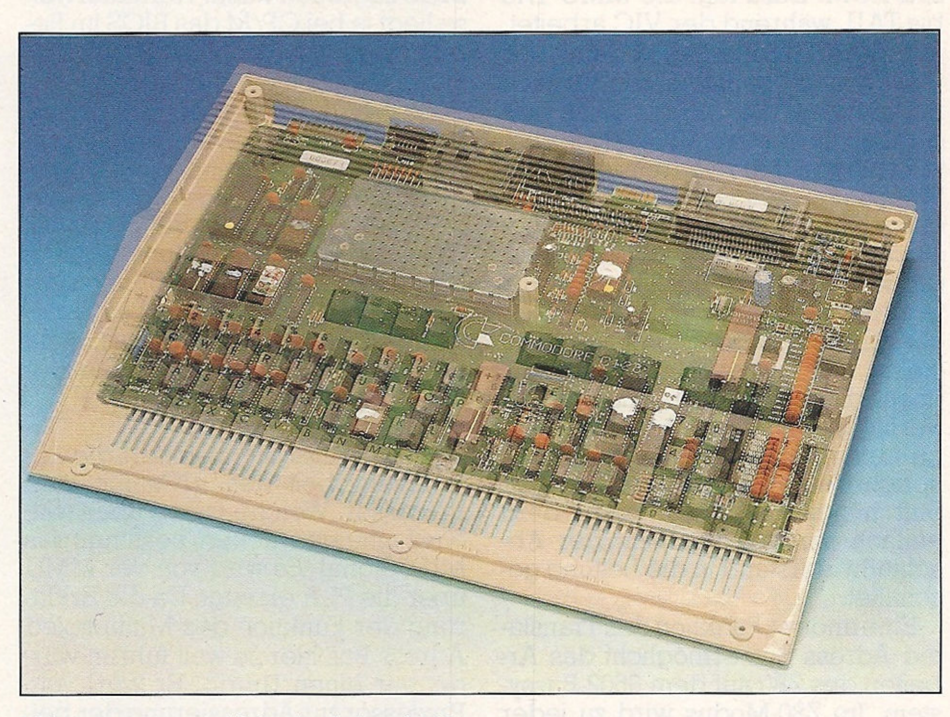


Bild 7. Ein »Röntgenblick« durch den C 128

Bild 1 zeigt die Platine des C 128. Rechts unten finden Sie die wichtigsten Bausteine der ganzen Platine: den 8502-Prozessor und den Z80. Der 8502 wurde von Commodore/MOS für den C 128 gebaut. In seinen Funktionen entspricht er dem 6510 aus dem C 64. Allerdings kann er mit 2 MHz getaktet werden, doppelt so schnell wie der 6510. Durch den Z80 »versteht« der C 128 auch CP/M-Programme. Sie können dadurch auch Programme wie Wordstar und Sprachcompiler (Pascal, Cobol und Ada) verwenden.

Rechts unten befindet sich das Gedächtnis des C 128: der 128-KByte-RAM-Speicher. 128 KByte RAM erhält man durch Parallelschaltung von sechzehn 8 KByte/8 Bit RAMs.

C 128-Hardware — Bewährtes und Neues

Gleich oberhalb der RAMs finden Sie die vier Betriebssystem-ROMs mit je 16 KByte. Im freien Steckplatz für ein 32-KByte-ROM wird in der deutschen C 128-Version ein deutscher Zeichensatz untergebracht sein. Sie können dort aber auch eigene Betriebssysteme einstecken.

Neben den ROMs, die Bildgeneratoren: Der RGB-Controller und der aus dem C 64 bekannte VIC, der für den C 128 leicht überarbeitet wurde. Normalerweise befindet sich über den beiden Video-Chips ein Abschirmblech. Ebenso über dem Modulator, der ein Hochfrequenzsignal zum Anschluß eines Fernsehers erzeugt. Im 2-MHz-Modus löst der RGB-Controller den VIC, der nur mit 1 MHz arbeiten kann, ab.

Rechts neben der »Bildstelle« befindet sich der Sound-Chip, der SID. Er ist identisch mit dem 6581 aus dem C 64. Etwas weiter rechts liegt die MMU, die Memory Management Unit. Erst mit der MMU kann ein 8502-Prozessor 128 KByte Speicher mittels Bank-Switching verwalten.

Die PLA (Programmable Logic Array, vom Hersteller programmierbare Logikschaltung) und die CIAs (Complex Interface Adapter, hier zur Steuerung von Aus- und Eingängen) haben im C 128 fast die gleichen Aufgaben wie im C 64. So verwaltet die PLA den ROM-Speicheraufbau. Sie selektiert die vom Prozessor benötigten ROMs im Computer und EPROMs auf einer eventuellen Erweiterungskarte. Die 6526-CIAs, Commodore/MOS-Entwicklungen, regeln alles, was an Daten in den C 128 hinein- und hinausgeht. CIA 1 übernimmt die Auswertung, welche Taste der Tastatur gedrückt wurde und teilt dem Prozessor mit, in welche Richtung ein Joystick weist. CIA 2 steuert die Ein- und Ausgänge des User-Ports und des seriellen IEC-Bus.

Bild 2 zeigt das Blockschaltbild des C 128. Im folgenden soll kurz der Aufbau des Systembusses erläutert werden.

1. Der Prozessor-Bus

Als Prozessorbus bezeichnet man den Adreß- und Datenbus. Der Adreßbus umfaßt 16 Leitungen A0 bis A15, der Datenbus 8, D0 bis D7. Wie Sie ja sicherlich wissen, muß beim C 64 ein Datum (POKE-Wert) innerhalb des Wertebereichs von 0 — 255 liegen. Aber nicht nur beim C 64, sondern bei allen 8-Bit-Computern, da 255 die größte mit 8 Bit darstellbare Zahl ist. Es sind deshalb 16 Adreßleitungen nötig, damit ein 8-Bit-Computer 65536 Speicherstellen (64 KByte) ansprechen kann. Beide Bussysteme stellen die Verbindung des Prozessors mit den ROMs, RAMs und I/O (Input/Output)-Bausteinen her. Der Prozessorbus dient sowohl dem 8502, als auch dem Z80. Damit der Z80, der mit 4 MHz getaktet wird, mit dem 2 MHz-Bussystem arbeiten kann, wurde ihm eine entsprechende Interface-Schaltung »verpaßt«. Diese Schaltung sorgt dafür, daß der Z80 nur mit 2 MHz getaktet wird, wenn er auf den Bus zugreift.

2. Der Translated Adress Bus

Dieses Bussystem wird von der MMU generiert. Das System besteht nur aus den höherwertigen Adreß-

leitungen TA8 bis TA15. Diese Adreßleitungen benötigt der VIC-Chip zum Zugriff auf das Zeichensatz-ROM. Dazu legt die MMU TA8 bis TA11, während der VIC arbeitet, auf einen Tri-State-Pegel, damit der VIC-Chip diese Leitungen für sich nutzen kann, als VA8 bis VA15. Das ermöglicht erst den VIC das gesamte Zeichen-ROM zu adressieren. Ein Zeichen besteht aus 8 Byte, der Zeichensatz umfaßt 256 Zeichen, insgesamt also 2048 Byte. Mit acht Leitungen könnte der VIC aber nur 256 Byte adressieren. Um aber 2048 Speicherplätze adressieren zu können, stellte eben die MMU dem VIC noch drei Tri-State-Leitungen zur Verfügung. Schaltet ein Baustein eine Leitung auf Tri-State-Pegel, entspricht das einer Leitung die in »der Luft hängt«. Technisch wird dazu einfach ein sehr hochohmiger Abschlußwiderstand in die Leitung geschaltet.

Eine andere Funktion des Translated Adress Bus ermöglicht das Arbeiten des Z80 auf dem 8502-Bussystem. Im Z80-Modus wird zu jeder Adresse von Bank 0 aus dem Bereich von $0000 bis $0FFF $D000 addiert, damit der Z80 auf das CP/M-BIOS zugreifen kann. Normalerweise liegt ja bei CP/M das BIOS im Bereich von $D000 bis $DFFF im RAM-Speicher. Da aber die ganze C 128-Organisation 8502-spezifisch ist und deshalb das BIOS in Bank 0 von $0000 bis $0FFF untergebracht ist, ist eine Konvertierung der Adressen von $D000 bis $DFFF auf $0000 bis $0FFF nötig. Und zwar immer dann, wenn vom Z80 auf das BIOS zugegriffen wird. Die Adreßkonvertierung findet nur in Bank 0 statt, nicht in Bank 1. Hier befindet sich die normale 8502-System-Zeropage.

Der Multiplexed Adress Bus

Dieses Bussystem sichert die Zusammenarbeit des VICs mit dem Prozessor. Welche Daten über welche Leitungen fließen, bestimmt das MUX-Signal. Es wird von der MMU über die PLA erzeugt. Da die Erklärung der Funktion des Multiplexed Adress Bus hier zu weit führen würde, nur einen Umriß: Er wird vom Prozessor zur Adressierung der beiden RAM-Bänke gebraucht, zur Adressierung der VIC-Register und für Zugriffe des VIC auf das Farb-RAM und das Zeichensatz-ROM.

Zum neuen Computer gibt es auch ein neues Floppy-Laufwerk, die 1571, die wir auch unter die Lupe nahmen. Zum Test standen leider keine Unterlagen zur Verfügung.

Das 1571-Laufwerk

Genaue Aussagen können wir über die Ladegeschwindigkeiten im C 128- und C 64-Modus geben. Im C 64-Modus hat sich nichts gegenüber der 1541 geändert, im 128er-Modus ist das Laufwerk allerdings 7- bis 8mal so schnell. So benötigt beispielsweise ein Hires-Bild (33 Blöcke auf Diskette) ganze 3,2 Sekunden Ladezeit. Der Bildschirm wird während des Ladens nicht abgeschaltet.

Auch das Formatieren geht sehr viel schneller als vorher: Ganze 40 Sekunden werden zum doppelseitigen Formatieren einer Diskette benötigt. Danach stehen 1328 Blöcke

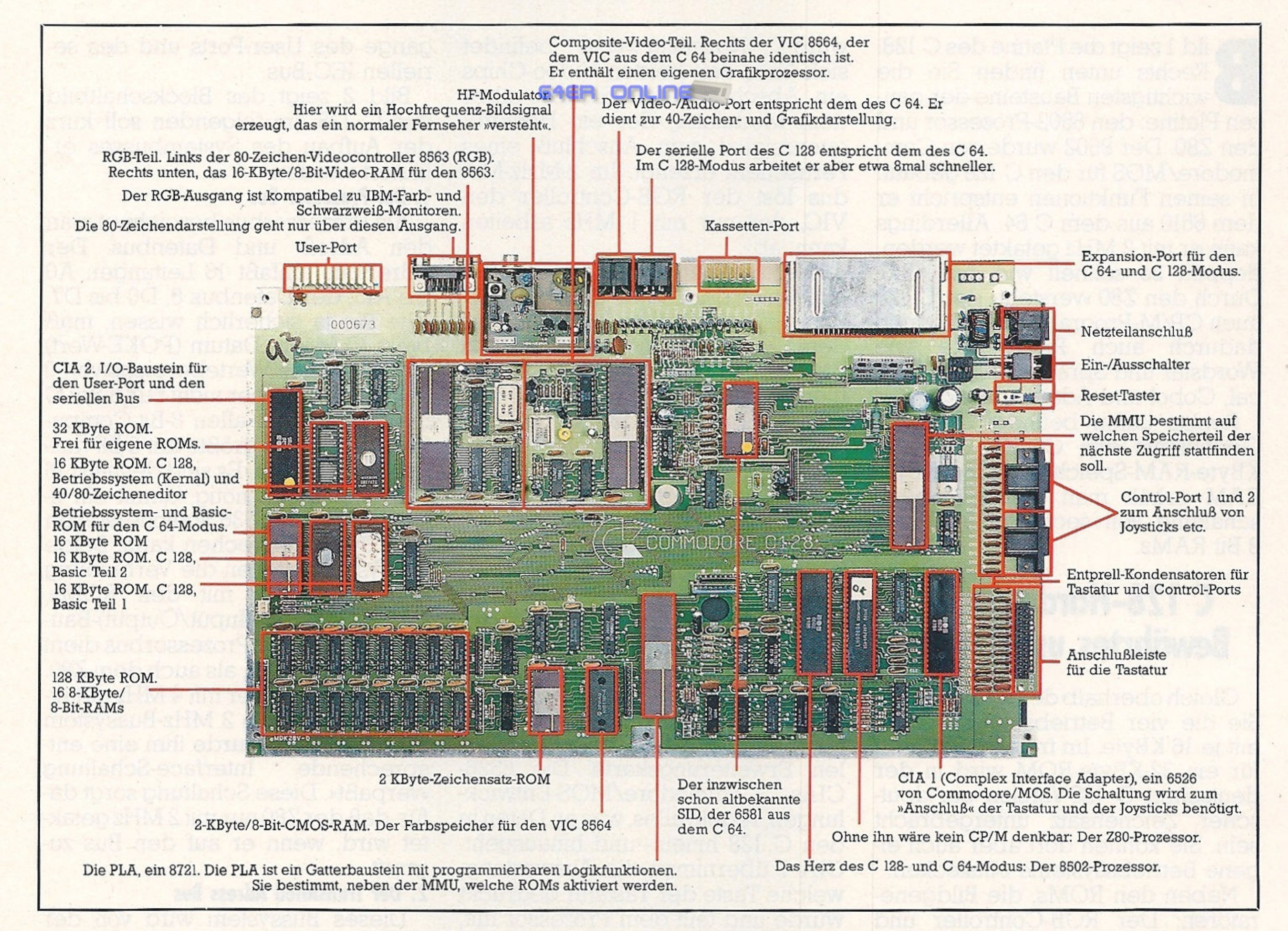


Bild 1. Die Platine des C 128. Das Abschirmblech des RGB- und Composite-Videoteils wurde abgenommen. Ebenso das des Modulators. Über der gesamten Platine liegt nochmals ein Abschirmdeckel, der auch als Kühlkörper für die ICs verwendet wird.

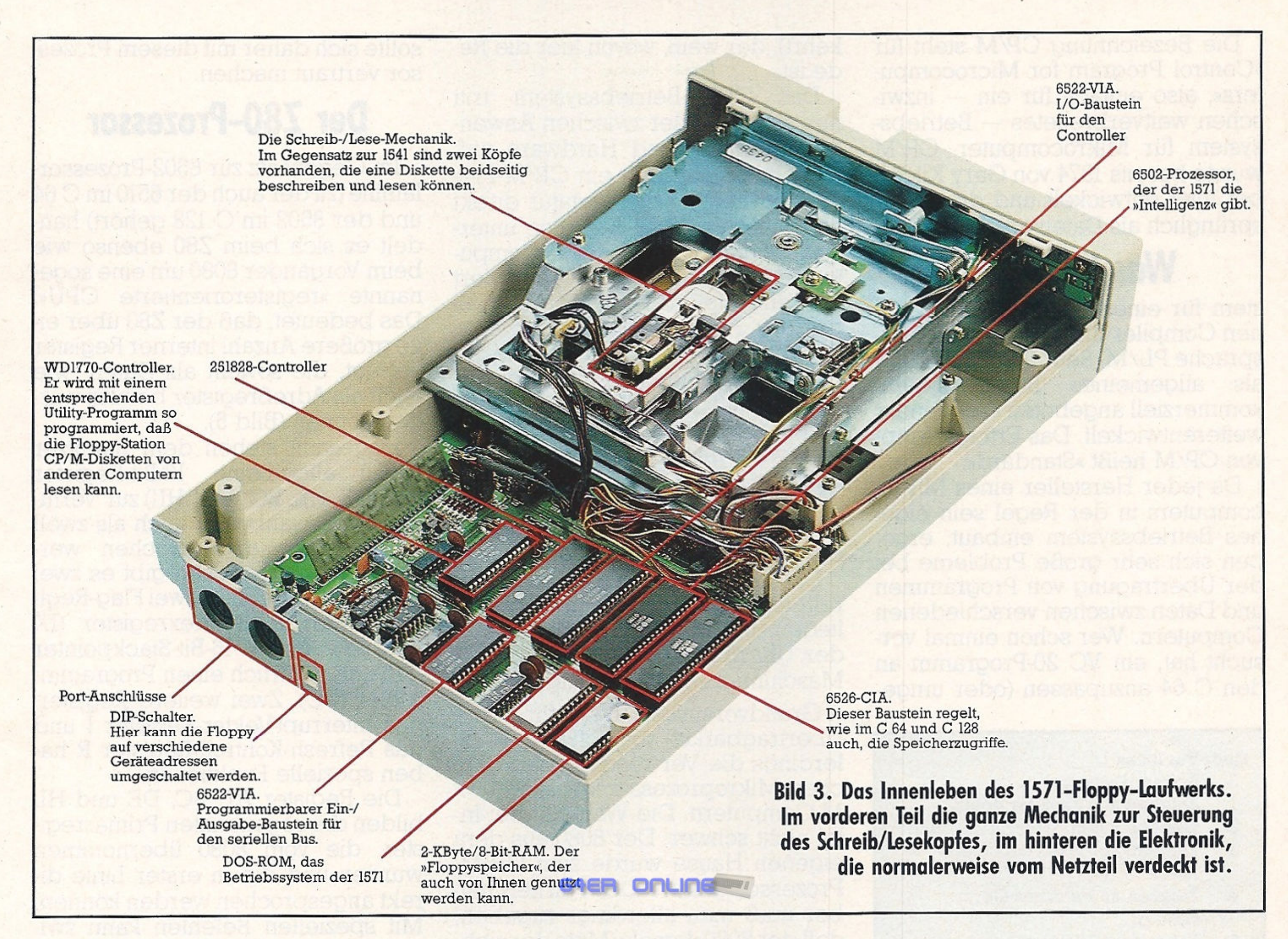


Bild 3. Das Innenleben des 1571-Floppy-Laufwerks. Im vorderen Teil die ganze Mechanik zur Steuerung des Schreib/Lesekopfes, im hinteren die Elektronik, die normalerweise vom Netzteil verdeckt ist.

zur freien Verfügung, also die doppelte Speicherkapazität einer 1541-formatierten Diskette. Noch eins fällt beim Formatieren auf: das Rattern entfällt. Den Grund dafür entdeckten wir nach dem Aufschrauben der 1571: Der mechanische Anschlag zur Kopfpositionierung wurde durch eine Lichtschranke ersetzt. Die Mechanik macht allgemein einen sehr guten und stabilen Eindruck. Interessant ist auch, daß eine weitere Lichtschranke zur Abfrage des Indexloches vorhanden ist. Alle Schreib-/Lesevorgänge funktionierten nämlich auch mit Disketten, die kein Indexloch hatten. Eine Ausnahme könnte der CP/M-Modus sein, wenn Disketten anderer Formate verwendet werden. Dies konnten wir aber mit dieser Version nicht testen, da der Floppy-Controller noch nicht auf andere Diskettenformate programmiert werden konnte.

Auch die Elektronik zeigt sich von einer guten Seite. Auf der sehr sauber aufgebauten Platine sind gegenüber der 1541 einige Chips hinzugekommen, über deren Funktion teilweise nur spekuliert werden konnte. So ist zum Beispiel eine 6526-CIA und ein WD1770-Controller vorhanden (Bild 3).

Das alte 1541-Problem der Wärmeabfuhr wurde bei der 1571 gelöst. Das ohnehin schon recht »kalte« Netzteil ist völlig von der Floppymechanik getrennt. Es können also kaum mehr Schreib-/Leseprobleme auftreten, die auf eine Wärmeleitung vom Netzteil auf die Floppymechanik zurückzuführen sind.

Wie der C 128 hat auch das Floppy-Laufwerk einen Modus, der ein früheres Commodore-Gerät, die 1541, simulieren soll. Hier gab's, zumindest bei unserer Vorabversion, noch einige Probleme. Es ist deshalb die Serienversion abzuwarten, will man ein endgültiges Urteil abgeben. Eine Kleinigkeit ist uns noch sehr positiv aufgefallen: An der Rückseite befinden sich zwei kleine DIP-Schalter, mit denen die Geräteadresse der Floppy auf Werte zwischen 8 und 11 geändert werden kann. Es wird also auch für absolute Anfänger ein problemloser Betrieb mit mehreren Laufwerken möglich sein.

Der CP/M-Modus

Der C 128 ist der erste Commodore-Computer, der von Haus aus bereits CP/M-fähig ist. Mit diesem Betriebssystem steht dem Anwender die ganze Welt der CP/M-Software zur Verfügung, was insbesondere beim mehr professionell orientierten Anwender ein starkes Argument für den C 128 darstellen dürfte. Allerdings wurde dieser Vorteil, wie beschrieben, mit einem immensen technischen Aufwand erkauft. Während ein reiner Z80-Computer mit CP/M in der Regel überhaupt keine Probleme hat (System in den Speicher laden — fertig), sind beim C 128 eine Vielzahl soft- und hardwaremäßiger Probleme zu lösen.

Bei dem uns zur Verfügung stehenden Testgerät war denn auch das CP/M-System noch nicht in der endgültigen Version implementiert. Insbesondere die Ladegeschwindigkeit unter CP/M ließ noch sehr zu wünschen übrig. Die Ladezeit für das CP/M-System selbst betrug in der (vorläufigen) Version ziemlich genau 45 Sekunden. In der endgültigen Version wird die Geschwindigkeit allerdings deutlich höher liegen, so daß ein sinnvolles Arbeiten möglich wird.

Wer bisher nur mit dem C 64 gearbeitet hat, dem stehen bei der Umstellung auf eine CP/M-fähige Maschine natürlich einige Umstellungen bevor.

Die Bezeichnung CP/M steht für »Control Program for Microcomputers«, also einfach für ein — inzwischen weitverbreitetes — Betriebssystem für Mikrocomputer. CP/M wurde bereits 1974 von Gary Kildall bei Intel entwickelt und diente ursprünglich als Dateiverwaltungs-System für einen von Intel vertriebenen Compiler für die Programmiersprache PL/M. Seit 1975 wird CP/M als allgemeines Betriebssystem kommerziell angeboten und ständig weiterentwickelt. Das Erfolgsrezept von CP/M heißt »Standard«.

Was ist CP/M?

Da jeder Hersteller eines Mikrocomputers in der Regel sein eigenes Betriebssystem einbaut, ergeben sich sehr große Probleme bei der Übertragung von Programmen und Daten zwischen verschiedenen Computern. Wer schon einmal versucht hat, ein VC 20-Programm an den C 64 anzupassen (oder umgekehrt), der weiß, wovon hier die Rede ist.

Das CP/M-Betriebssystem tritt nun als Vermittler zwischen Anwenderprogramm und Hardware auf. Beispielsweise fragt ein CP/M-Programm niemals die Tastatur direkt ab, denn die kann ja infolge unterschiedlicher Hardware von Computer zu Computer anders konzipiert und angesteuert sein. Statt dessen ruft das Anwenderprogramm eine Routine im CP/M-System auf, die die gewünschte Funktion ausführt. Natürlich muß das CP/M-System für jeden Computer an die spezielle Hardware angepaßt werden, aber, und das ist das Entscheidende, die Anwenderprogramme laufen unverändert, da sie ja nicht direkt auf die Hardware zugreifen.

Da alle CP/M-Programme generell im gleichen Speicherbereich liegen, treten kaum Probleme bei der Übertragung von einer CP/M-Maschine zur anderen auf.

Grundvoraussetzung für die Übertragbarkeit von Software ist allerdings die Verwendung des gleichen Mikroprozessors in allen CP/M-Computern. Die Wahl fiel bei Intel nicht schwer: Der 8080 aus dem eigenen Hause wurde zum CP/M-Prozessor gekürt. Der Fairneß halber muß man allerdings zugeben, daß der 8080 damals (Mitte der siebziger Jahre) tatsächlich einer der leistungsfähigsten Mikroprozessoren überhaupt war.

Heutzutage wird allerdings fast ausschließlich der Z80 verwendet, der voll aufwärtskompatibel zum 8080 ist, aber über einen stark erweiterten Befehlsvorrat und mehr interne Register verfügt. Wer sich ernsthaft für CP/M interessiert, der sollte sich daher mit diesem Prozessor vertraut machen.

Der Z80-Prozessor

Im Gegensatz zur 6502-Prozessorfamilie (zu der auch der 6510 im C 64 und der 8502 im C 128 gehört) handelt es sich beim Z80 ebenso wie beim Vorgänger 8080 um eine sogenannte »registerorientierte CPU«. Das bedeutet, daß der Z80 über eine größere Anzahl interner Register verfügt, die sowohl als Daten- wie auch als Adreßregister benutzt werden können (Bild 5).

Insgesamt stehen dem Benutzer sechs allgemeine 16-Bit-Register (BC, DE, HL, BC', DE', HL') zur Verfügung, die wahlweise auch als zwölf 8-Bit-Register angesprochen werden können. Daneben gibt es zwei Akkus und ebenfalls zwei Flag-Register, zwei 16-Bit-Indexregister (IX und IY), einen 16-Bit-Stackpointer (SP) und natürlich einen Programmzähler (PC). Zwei weitere Register, das Interrupt-Vektor-Register I und das Refresh-Kontroll-Register R haben spezielle Funktionen.

Die Register AF, BC, DE und HL bilden die sogenannten Primärregister, die vom 8080 übernommen wurden und die in erster Linie direkt angesprochen werden können. Mit speziellen Befehlen kann zwischen diesen Primärregistern und den »Sekundärregistern« AF', BC', DE' und HL' umgeschaltet werden. Die Indexregister IX und IY haben ähnliche Aufgaben wie die 6502-Register X und Y, sind jedoch 16 Bit breit.

Der Z80-Befehlssatz ist sehr umfangreich und umfaßt unter anderem auch 16-Bit-Arithmetik, Blockverschiebebefehle, automatische

Code	Funktion
0	System-Kaltstart
1	Zeichen von Tastatur holen
2	Zeichen auf Bildschirm ausgeben
3	Zeichen von externem Gerät holen
4	Zeichen an externes Gerät senden
5	Zeichen an Drucker senden
6	Direkte Ein-/Ausgabe über Konsole
7	I/O-Zuordnung holen
8	I/O-Zuordnung festlegen
9	String auf Bildschirm ausgeben
10	String von Tastatur einlesen
11	Feststellen, ob Taste gedrückt
12	Liefert Versionsnummer des CP/M-Systems
13	Reset des Diskettensystems
14	Laufwerk auswählen
15	Datei öffnen
16	Datei schließen
17	Datei auf Diskette suchen
18	Zweite Datei suchen
19	Datei löschen
20	Nächsten Block lesen
21	Nächsten Block speichern
22	Neue Datei erzeugen
23	Dateinamen ändern
24	Angesprochene Laufwerke feststellen
25	Aktuelles Laufwerk feststellen
26	DMA-Pufferadresse festlegen
27	Bit-Map-Adresse holen
28	Schreibschutz setzen
29	Schreibschutzinformation holen
30	Datei-Attribute setzen
31	File-Parameter-Adresse holen
32	USER-Nummer setzen
33	Daten von Diskette lesen
34	Daten auf Diskette schreiben
35	Dateigröße feststellen
36	Datensatzadresse holen

Tabelle 1. Die CP/M-Funktionsaufrufe



Bild 4. Die Tastatur ist funktionell und ergonomisch aufgebaut. Der deutsche Zeichensatz wird durch die Umschalttaste ASCII/DIN erreichbar. Die Belegung der Tasten entspricht dann den DIN-Normen. Die Bezeichnungen sind bereits auf den Tastenkappen aufgedruckt. Durch diesen Zeichensatz ergibt sich teilweise eine fünffache Belegung einzelner Tasten. Doch diese Tatsache wird dann gerechtfertigt, wenn es speziell auf den deutschen Markt zugeschnittene Textverarbeitungsprogramme geben wird (und vielleicht auch einen Drucker von Commodore, der die deutschen Umlaute von Haus aus beherrscht).

Such- und Ein-/Ausgabebefehle, bedingte Unterprogrammaufrufe sowie Einzelbitbefehle. Insgesamt kennt die Z80-CPU über 700 Opcodes. Um diese vielen Maschinenbefehle verschlüsseln zu können (mit einem Befehlsbyte können nur 256 Befehle codiert werden), gibt es einige spezielle »Umschaltopcodes«, die einfach bewirken, daß der Z80 intern auf eine andere Befehlstabelle umschaltet und das nächste Befehlsbyte in einer anderen Form interpretiert.

Nach diesem kurzen Ausflug zum Z80 jetzt jedoch wieder zum CP/M-Betriebssystem. Beim Studium von Handbüchern und CP/M-Literatur stößt man immer wieder auf vier Abkürzungen: BIOS, BDOS, CCP und TPA. Hinter diesen Bezeichnungen verbergen sich die wichtigsten Bestandteile eines CP/M-Systems.

BIOS steht für »Basic Input Output System«, hat aber absolut nichts mit der Programmiersprache Basic zu tun. Die Bezeichnung deutet lediglich an, daß es sich hierbei um eine Sammlung grundlegender Routinen zur Ein-/Ausgabe von Daten handelt.

Das CP/M-BIOS entspricht von der Funktion her ziemlich genau den Kernal-Routinen bei Commodore-Computern. Der gesamte Kontakt eines Anwenderprogramms mit der Hardware-Umgebung läuft über diese Routinen. Alle BIOS-Routinen werden indirekt über das BDOS, das Basic Disk Operating System, durchgeführt. Der Aufruf aller Systemroutinen geschieht durch einen Unterprogrammsprung zur Adresse 5, dem sogenannten BDOS-CALL. Die verschiedenen geforderten Funktionen werden dadurch selektiert, daß im C-Register des Z80 ein Funktionscode übergeben wird. Die Routine ab Adresse 5 verzweigt dann abhängig vom Inhalt des C-Registers zu den verschiedenen BIOS-Funktionen (Tabelle 1).

Diese beiden Teile des CP/M-Systems, BIOS und BDOS, werden beim »Booten«, also beim Laden des Systems von der Diskette, einmal in den Speicher geholt und bleiben dann ständig resident vorhanden.

Für den Kontakt zwischen CP/M und Anwender sorgt der »Console Command Processor« (CCP). Es handelt sich dabei um einen einfachen Kommando-Interpreter, der über die Tastatur eingegebene CP/M-Kommandos erkennt und ausführt. Mit CP/M-Kommandos hat es dabei eine besondere, grundlegende Bewandtnis. Hierzu ist es wichtig zu wissen, daß Dateinamen unter CP/M aus zwei Teilen bestehen, der eigentlichen Dateibezeichnung (acht Zeichen Länge) und der Namenserweiterung (»Extension«, drei Zeichen Länge). Diese Erweiterung, die durch einen Punkt vom eigentlichen Namen abgetrennt wird, gibt den Typ der Datei an.

Kommandostruktur

Besonders wichtig ist der Dateityp »COM«. Wenn ein Dateiname die Erweiterung COM hat, dann stellt sie ein CP/M-Kommando dar. Das funktioniert folgendermaßen: Wenn der Benutzer ein Kommando eingibt, dann sucht der CCP auf der Diskette nach einer COM-Datei, die den Namen des Kommandos trägt. Diese Datei wird dann geladen und gestartet, mit anderen Worten, die Datei **ist** das Kommando.

Will man beispielsweise das Directory der Diskette sehen, dann gibt man den Befehl DIR ein. Der CCP sucht, lädt und startet daraufhin das Programm »DIR.COM«, das wiederum das Inhaltsverzeichnis der Diskette ausgibt. Derartige Kommandos heißen »transiente Kommandos«, weil sie eben nicht fest eingebaut sind, sondern nur bei Bedarf geladen werden. Der für transiente Kommandos reservierte Speicherbereich heißt »Transient Program Area« (TPA), womit auch

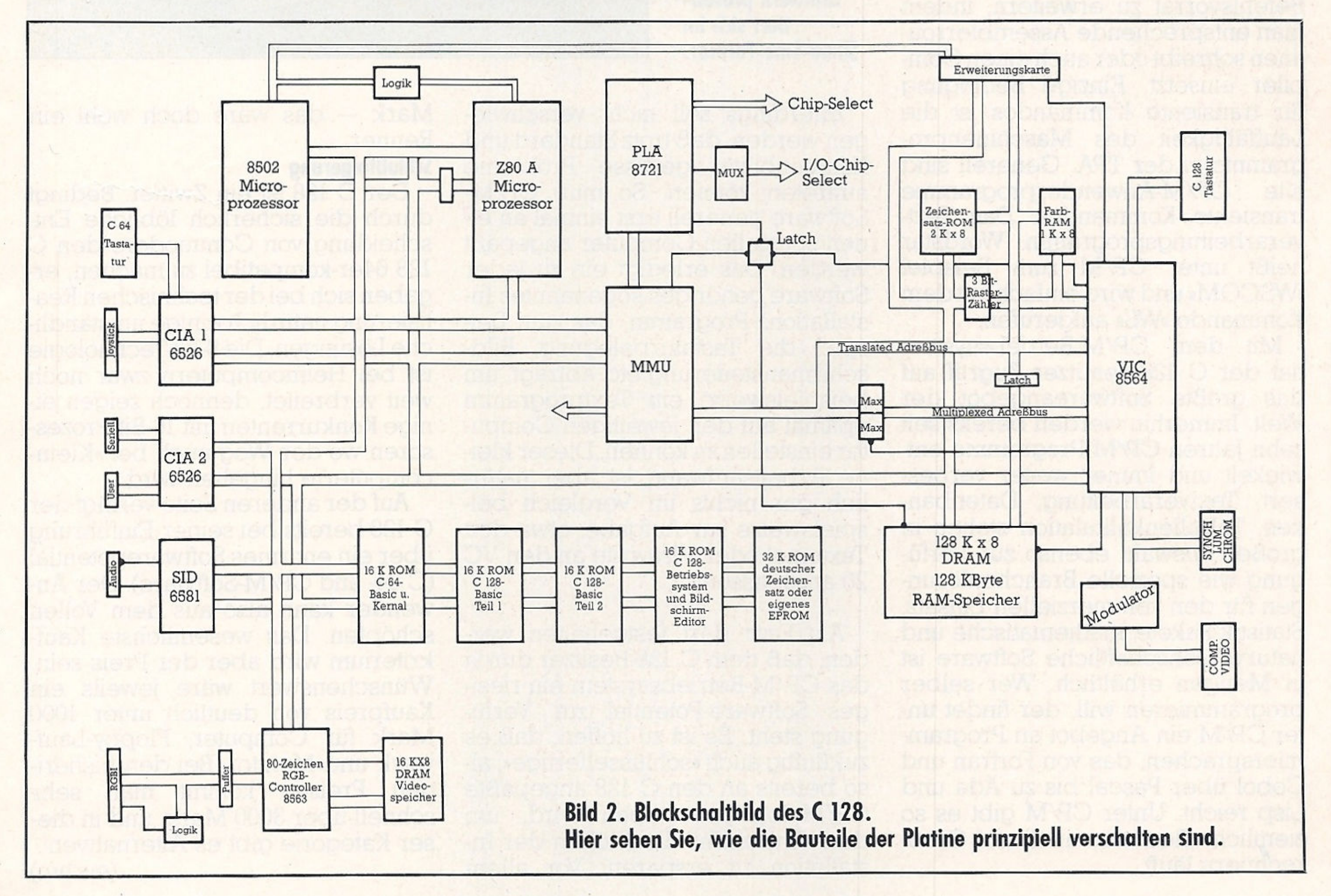


Bild 2. Blockschaltbild des C 128. Hier sehen Sie, wie die Bauteile der Platine prinzipiell verschalten sind.

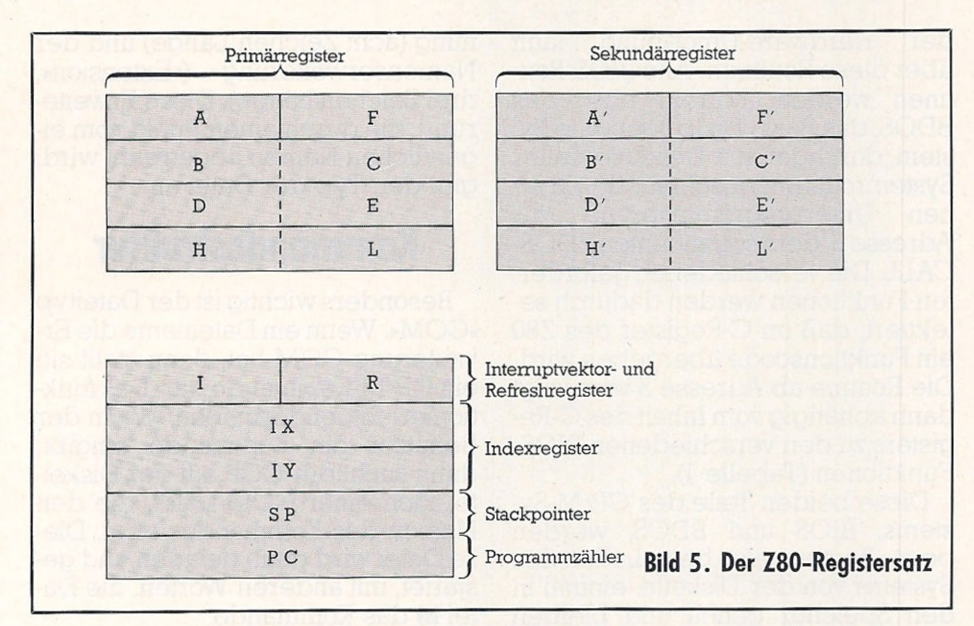


Bild 5. Der Z80-Registersatz

die vierte der oben angesprochenen Abkürzungen geklärt wäre. Jedes CP/M-System verfügt auf der Systemdiskette über eine Standard-Bibliothek von wichtigen transienten Kommandos. Dazu gehören in der Regel Kommandos zum Ändern von CP/M-Parametern, zum Kopieren, Löschen, Listen und Umbenennen von Dateien und ähnliches, was man zum sinnvollen Arbeiten mit einer Diskettenstation benötigt.

Natürlich ist es möglich, diesen Befehlsvorrat zu erweitern, indem man entsprechende Assemblerroutinen schreibt oder auch einen Compiler einsetzt. Einzige Bedingung für transiente Kommandos ist die Lauffähigkeit des Maschinenprogramms in der TPA. Generell sind alle CP/M-Anwenderprogramme transiente Kommandos. Das Textverarbeitungsprogramm Wordstar heißt unter CP/M zum Beispiel »WS.COM« und wird einfach mit dem Kommando »WS« aufgerufen.

Mit dem CP/M-Betriebssystem hat der C 128-Benutzer Zugriff auf das größte Softwareangebot der Welt. Immerhin werden bereits seit zehn Jahren CP/M-Programme entwickelt und immer weiter verbessert. Textverarbeitung, Datenbanken, Tabellenkalkulation stehen in großer Auswahl ebenso zur Verfügung wie spezielle Branchenlösungen für den kommerziellen Einsatz. Statistik-Pakete, mathematische und naturwissenschaftliche Software ist in Mengen erhältlich. Wer selber programmieren will, der findet unter CP/M ein Angebot an Programmiersprachen, das von Fortran und Cobol über Pascal bis zu Ada und Lisp reicht. Unter CP/M gibt es so ziemlich alles, was auch auf Großrechnern läuft.



Bild 6. Das neue 1571-Laufwerk präsentiert sich im Slime-Line-Format.

Allerdings soll nicht verschwiegen werden, daß trotz Standard und Kompatibilität gewisse Probleme auftreten können. So muß CP/M-Software generell erst einmal an einen speziellen Computer angepaßt werden. Das erledigt ein zu jeder Software gehöriges sogenanntes Installations-Programm, das zum Beispiel die Tastaturbelegung, Bildschirmansteuerung etc. abfragt, um beispielsweise ein Textprogramm optimal auf den jeweiligen Computer einstellen zu können. Dieser kleine Arbeitsaufwand ist aber natürlich gar nichts im Vergleich beispielsweise zur Aufgabe, etwa den Textomat oder Vizawrite an den VC 20 anzupassen.

Als Fazit darf festgehalten werden, daß dem C 128-Besitzer durch das CP/M-Betriebssystem ein riesiges Software-Potential zur Verfügung steht. Es ist zu hoffen, daß es zukünftig auch »schlüsselfertige«, also bereits an den C 128 angepaßte CP/M-Software geben wird, um dem Anwender die Mühen der Installation zu ersparen. Vor allem aber dürfte es von entscheidender Bedeutung sein, ob die Hersteller von CP/M-Software willens sind, die Preise für ihre Produkte, die sich teilweise noch im vierstelligen Bereich befinden, drastisch zu senken.

Die Chancen dafür stehen nicht schlecht, denn nachdem CP/M im professionellen Bereich inzwischen durch MS-DOS, ein 16-Bit-Betriebssystem, abgelöst worden ist, sollten die CP/M-Hersteller eigentlich schon an neuen Märkten interessiert sein, und diese neuen Märkte könnten durchaus im oberen Home-Bereich liegen. Wenn die Software jedoch teurer als der Computer ist, dürfte wohl kein Geschäft zu machen sein, aber beispielsweise Wordstar für 298 oder gar für 198 Mark — das wäre doch wohl ein Renner.

Schlußfolgerung

Der C 128 ist ein Zwitter. Bedingt durch die sicherlich löbliche Entscheidung von Commodore den C 128 64er-kompatibel zu machen, ergaben sich bei der technischen Realisierung natürlich einige umständliche Lösungen. Die 8-Bit-Technologie ist bei Heimcomputern zwar noch weit verbreitet, dennoch zeigen einige Konkurrenten mit 16-Bit-Prozessoren wo der Weg auch bei »Kleincomputern« hingehen wird.

Auf der anderen Seite verfügt der C 128 bereits bei seiner Einführung über ein enormes Softwarepotential (C 64- und CP/M-Software). Der Anwender kann also aus dem Vollen schöpfen. Das wesentlichste Kaufkriterium wird aber der Preis sein. Wünschenswert wäre jeweils ein Kaufpreis von deutlich unter 1000 Mark für Computer, Floppy-Laufwerk und Monitor. Bei den bisherigen Preisen kommt man sehr schnell über 3000 Mark, und in dieser Kategorie gibt es Alternativen!

(ev/hm)


64er online

Die Alternativen

Drucken tun Sie alle. Nur auf das »Wie« kommt es an. An dieser Stelle erklären wir Ihnen, wie Thermodrucker, Tintenstrahldrucker und Plotter funktionieren und wozu man Sie verwendet.

Sowohl Tintenstrahl- und Thermodrucker haben mit den Nadel-Druckern eines gemeinsam: den Aufbau von Zeichen aus einzelnen Punkten. Beim Druck wird immer eine senkrechte Punktreihe (5-9 Punkte) auf einmal zu Papier gebracht. Dann rückt der Druckkopf ein paar 10tel Millimeter weiter und setzt die nächste Punktreihe. So wird Stück für Stück jedes Zeichen zusammengesetzt. Die Ansteuerung des Druckkopfes übernimmt eine Elektronik im Drucker, die aus den ASCII-Codes die Punktmatrix berechnet.

Wie funktioniert ein Thermodrucker?

Schon der Name gibt Auskunft über das Funktionsprinzip. Ob nun Thermodrucker, Thermostat oder Thermosflasche, alles was mit »Thermo« anfängt, hat in den meisten Fällen etwas mit Hitze zu tun.

Mit Wärme drucken

Wie kann man Hitze drucken? Nun, hierzu ist ein spezielles Papier nötig, das sich schwarz verfärbt wenn es heiß wird. Das Papier stellt man durch Beschichtung mit chemischen Substanzen her, die bei Wärmezufuhr zu einer schwarzen Verbindung reagieren. Erkennen kann man dieses Papier an seiner einseitig sehr glatten Oberfläche. Nicht zu verwechseln mit dem silbrig beschichteten Papier einiger »Kurzschluß-Drucker«, welche die metallische Beschichtung beim Druck wegschweißen.

Anstelle der Nadeln hat der Druckkopf eines Thermodruckers kleine, punktförmige Plättchen die schnell aufgeheizt werden können und auch schnell wieder abkühlen. Bild 1 zeigt die schematische Anordnung der Heizplättchen in einem Thermodruckkopf. Die Heizplättchen stellen im Prinzip nichts weiter dar, als flache »Widerstände«, die senkrecht übereinander angeordnet auf einer gemeinsamen Grundplatte sitzen. Wenn bei einem Nadeldrucker bestimmte Nadeln aufs Papier geschlagen werden, erhitzt ein Thermodruckkopf einfach die entsprechenden Stellen. Das Ergebnis ist in beiden Fällen gleich: ein Muster aus schwarzen Punkten.

Allerdings ist nur das Ergebnis gleich. Die Druckgeschwindigkeit eines Thermodruckers ist weitaus geringer als die eines Impact-Matrixdruckers, wie man einen Nadeldrucker auch bezeichnet. Dafür ist beim Thermodruck fast nichts zu hören. Die eingeschränkte Geschwindigkeit eines Thermodruckers ergibt sich einfach aus der Aufheiz- und Abkühlzeit der Widerstandsplatten. Der Widerstand kann einfach nicht schlagartig aufgeheizt werden, da bei zu hohem Heizstrom die Widerstände einfach durchbrennen würden.

Teueres Papier

Sie können sich sicherlich vorstellen, daß das Spezialpapier wesentlich mehr kostet als normales »Computerpapier«. Die Thermodrucker-Hersteller haben sich deshalb noch etwas Zusätzliches einfallen lassen, damit auch normales Papier verwendet werden kann: Den Druck über ein Carbon-Band. Anstatt das Papier »schwarz zu brennen«, werden hier schwarze Partikel vom Carbon-Band gelöst und aufs Papier übertragen. Also ähnlich einer Schreibmaschine, nur, daß das Band nicht durch einen Anschlag abfärbt, sondern durch Hitzeeinwirkung. Aber das Carbon-Band ist wesentlich teuerer als normales Druckerband und kann nur einmal verwendet werden. Aus dem Druckprinzip folgt, daß keine Durchschläge gemacht werden können.

Tintenstrahldrucker — Wie funktionieren Sie?

Tintenstrahldrucker haben ein raffiniertes, mechanisch sehr aufwendiges Funktionsprinzip, das ein gutes Schriftbild (Bild 2) und eine hohe Druckgeschwindigkeit bei sehr leisem Druck ermöglicht. Sie vereinigen also alle Vorteile von Impact-Matrixdruckern und Thermodruckern in sich. Der einzige Nachteil: der Preis. Wesentlich unter 2000 Mark ist fast kein Drucker dieser Klasse zu haben.

Im Gegensatz zu allen anderen Druckern arbeiten diese Druckertypen nicht mit Farbbändern, sondern mit Tinte die gezielt aufs Papier gespritzt wird. Nur tröpfchenweise versteht sich. Damit die Tropfen schnell antrocknen und nicht verwischen, ist ein saugfähiges Papier Voraussetzung. Dieses Spezialpapier ist et-

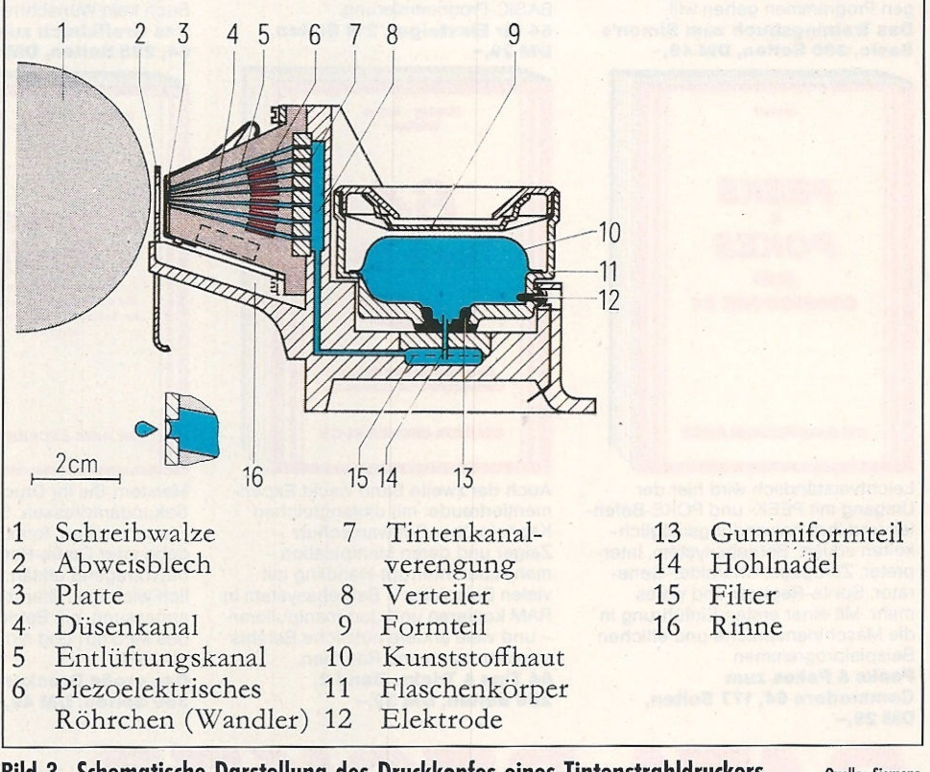


Bild 3. Schematische Darstellung des Druckkopfes eines Tintenstrahldruckers Quelle: Siemens

wa anderthalb mal so teuer wie normales Computerpapier. Es kann jedoch leicht beschafft werden.

So wie bei allen anderen Matrixdruckern auch, wird immer eine senkrechte Punktreihe in einem Zug geschrieben. Sehr interessant ist das Prinzip, mit dem die Tropfen auf Papier »geschossen« werden. Bild 3 zeigt einen Druckkopf schematisch.

Für jeden Punkt auf einer senkrechten Reihe der Zeichenmatrix gibt es einen Düsenkanal (4) zur Erzeugung der Tropfen. In der Zeichnung (Kopf des Siemens PT88/89) sieht man eine Reihe von sechs Düsen. Der gezeigte Kopf des Siemens PT 88/89 hat zwei solche Reihen versetzt zueinander angeordnet. Den Abschluß der Kanülen bildet eine Platte (3), die etwa 1 mm vom Papier entfernt ist. Jeder Düsenkanal ist konzentrisch von einem Piezokeramik-Röhrchen umschlossen. Legt man eine Spannung an diese Röhrchen an, ziehen sie sich zusammen und üben so einen kleinen »Schlag« auf die Flüssigkeitssäule aus. Durch den Druckanstieg wird dann ein Tröpfchen aus der Düse geschleudert. Wie sich der Tropfen bildet, sehen Sie in Bild 4.

Piezo-elektrische Kristalle haben die Eigenschaft, ihre Form in eine bestimmte Richtung zu verändern wenn eine Spannung angelegt wird. Umgekehrt liefert der Kristall einen sehr kleinen Stromstoß, wenn er gebogen wird (Piezo-Feuerzeug, Kristall-Tonabnehmer).

Bei diesem Verfahren liegt der Tintenvorratsbehälter (10, Bild 3) tiefer als die Kanülen. Das hat den Vorteil, daß in den Düsen ein statischer Unterdruck herrscht. Es kann also nur dann Tinte austreten, wenn sich das Piezoröhrchen zusammenzieht.

Das ist die übliche und verbreitetste Version des Tintenstrahldruckers. Von Hersteller zu Hersteller gibt es dabei natürlich Abweichungen. So kann der Druck auch durch schlagartiges Aufheizen von etwas Tinte bis zur Dampfbildung erzeugt werden. Innerhalb eines kleinen Behälters wird mit Hilfe einer leistungsstarken Heizspirale die Tinte in der Umgebung der Heizspirale in sehr kurzer Zeit zum Sieden gebracht, wobei sich Dampfbläschen bilden. Diese Dampfentwicklung erhöht den Systemdruck und treibt einen Tropfen Tinte aus der Düse. Durch das Verdampfen bilden sich leicht Feststoff-Rückstände, die den Heizbehälter schnell verstopfen können.

Verschiedene Entwicklungen gehen in eine andere Richtung. Die

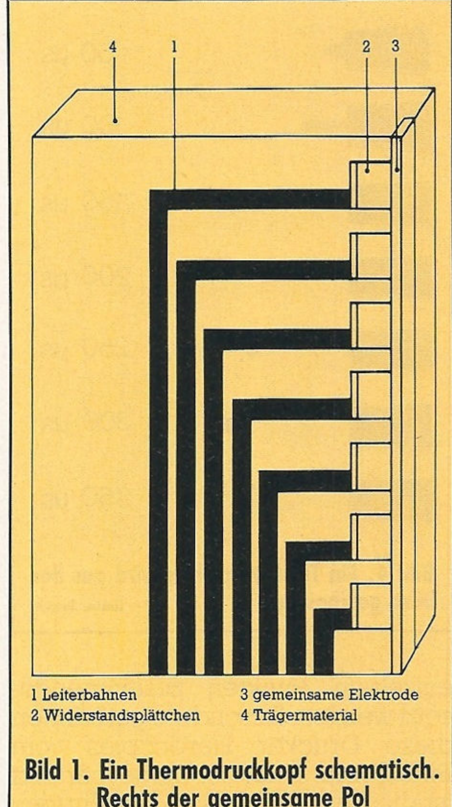


Bild 1. Ein Thermodruckkopf schematisch. Rechts der gemeinsame Pol der Widerstände

einzelnen Matrixpunkte werden nicht durch einzelne Kanülen erzeugt, sondern aus einer gemeinsamen Quelle. Die Tintentröpfchen verlassen dabei statisch aufgeladen die Piezo-»Pumpe« und werden über ein elektrisches Feld (Plattenkondensator) auf die richtige Bahn gebracht.

Da bei beiden Systemen nur sehr kleine Massen (Tintentröpfchen) bewegt werden müssen, können größere Schreibgeschwindigkeiten als bei Impact-Nadeldruckern erreicht werden.

Die Probleme die bei Tintenstrahldruckern gelöst werden müssen sind recht vielseitig. Das größte davon ist die Größe der Tintentröpfchen, die konstant sein muß. Bei ungleichen Tropfen wäre eine fleckige Schrift die Folge. Es ist also sehr wichtig, daß einmal die Kanülen genau gleichen Durchmesser haben und die Piezo-Kristalle den gleichen Druck auf die Flüssigkeitssäule erzeugen.

Recht arbeitsintensiv kann im Falle einer verstopften Düse das Reinigen des Druckkopfes werden. Der Druckkopf muß entweder ausgetauscht oder mit einer Reinigungsflüssigkeit gespült werden. Schmutzige Hände sind nach dieser Arbeit nicht selten. Beim Unterdruckverfahren, wo der Vorratsbehälter unterhalb der Düsen liegt, sind verstopfte Kanülen relativ selten. Häufiger kommen sie vor, wenn der Vorratsbehälter oberhalb angebracht ist. Die Tintensäule reicht dann nämlich bis an die Düsenöffnung und kann leicht antrocknen.

Plotter — was können Sie?

Müssen Sie häufig Kurvendiagramme zeichnen? Dann ist ein Plotter das richtige Druckgerät für Sie.

X-Y-Plotter

Diese Schreiber, auch Flachbett-Plotter genannt, eignen sich vor allem für kleinere Papiergrößen bis DIN A2. Das Papier wird auf einer Art Zeichenbrett festgeklemmt und ein Stift (Tinte oder Filzstift) über einen beweglichen Arm, der über die gesamte Papierbreite reicht, in Längsrichtung geführt. Über eine Seilzugmechanik kann der Stift in der Breite auf dem Arm hin- und herbewegt werden. Auf diese Weise kann jeder Punkt des Papiers erreicht werden. Eine Hubmechanik am Schreibkopf hebt den Stift an, wenn nicht geschrieben werden soll.

Die Positionierung des Stiftes erfolgt, auf etwa 0,1 mm (beim 1520-Plotter 0,2 mm) genau, über zwei Schrittmotoren, die über eine Treiberelektronik angesteuert werden. Computerseitig muß in den meisten Fällen nur die x- und y-Koordinate an den Plotter übertragen werden.

Trommel-Plotter

Sie sind für große Papierformate konstruiert worden, wo ein Flachbettplotter zuviel Platz in Anspruch nehmen würde. Anstelle einen Stift in x-Richtung mit einem Arm zu bewegen, bewegt man hier das Papier in Längsrichtung unter dem Schreibstift hin und her. Es sind dadurch keine Grenzen in der Papierlänge gesetzt. Eine Gummiwalze transportiert das Papier so, daß es auf beiden Seiten aus dem Plotter »hinaushängen« kann. Um den Schlupf möglichst gering zu halten, sind auf beiden Seiten der Walze kreisförmig Nadeln angeordnet, die zusätzlich das Papier mit hin- und herbewegen.

Die zeichnerischen Vorteile eines Plotters liegen auf der Hand. Linien werden durchgehend gezeichnet. Im Gegensatz zu Hardcopies bei Matrixdruckern, bei denen Linien aus Punkten zusammengesetzt sind und dadurch etwas »zerissen« aussehen. Meistens werden Plotter zum Zeichnen von Schalt- und Bauplänen verwendet. Die Erfassung und Bearbeitung erfolgt meist mit einem CAD (Computer Aided Design)-System.

Des Preises wegen sind sie im Heimbereich kaum anzutreffen. Flachbettplotter kosten in der Regel über 2000 Mark.

Welcher Typ paßt zu mir?

Zusammen mit den letzten beiden Ausgaben, haben wir Ihnen fast alle Drucker vorgestellt, die es gibt. Teure Exemplare über 2000 Mark haben wir mit Absicht nicht berücksichtigt. Der Preis paßt einfach nicht mehr zu einem Heimcomputer.

Thermodrucker gehören zu den billigsten Druckern, die es gibt. Allerdings nur in der Anschaffung. Das spezielle Thermopapier macht nämlich diese Drucker auf Dauer gesehen teuer. Auch wenn Sie auf Plastikcarbon-Band ausweichen, mit dem verschiedene Thermodrucker auch normales Papier verarbeiten, fallen hohe Gebrauchskosten an.

Durch das kaum hörbare Druckgeräusch sind diese Drucker besonders für »Nachtarbeiter« geeignet, die sich keinen Tintenstrahldrucker leisten wollen. Mit Thermodruckern können Sie Listings und Hardcopies (nicht bei allen Typen) drucken. Zum Schreiben von Briefen eignen sich die meisten wegen der schlechten Schrift (5x7 Punkte) nicht.

Empfehlen können wir diese Drucker denjenigen, die nicht zu tief in das Computer-Hobby einsteigen wollen.

Nadel-Matrixdrucker sind sehr weit verbreitet. Das liegt an der Vielsseitigkeit dieser Geräte. Es fällt im Heimbereich kaum ein Druckproblem an, das diese Drucker nicht bewältigen würden. Wenn Sie die Marktübersicht Matrixdrucker im 64'er 5/85 anschauen, sehen Sie, daß diese Drucker häufig über die verschiedensten Schriftarten verfügen. Sie können also sowohl Ihre Geschäftspost als auch Listings damit drucken. Die Korrespondenz in Schönschrift (etwa 50 Z/s) und die Listings in Normalschrift (etwa 100 Z/s), wobei die Normalschrift schon ein sehr gutes Schriftbild haben kann.

Die Vielseitigen

Nadel-Matrixdrucker können jedem wärmstens empfohlen werden.

Aber mit dem Schreiben von Briefen sind die Leistungen eines Matrixdrucker nicht erschöpft. Da die Nadeln (bei den meisten Matrixdruckern) einzeln angesteuert werden können, kann jede Grafik aus

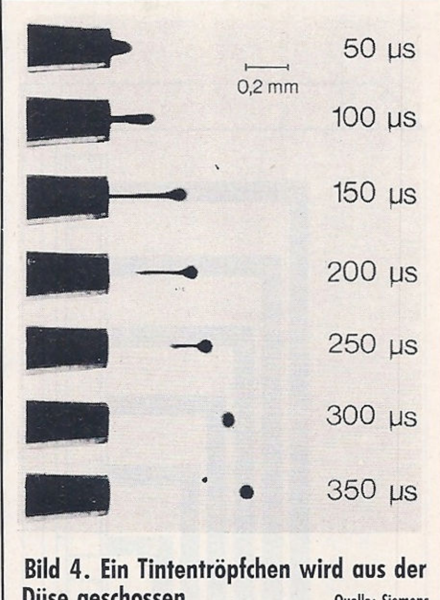


Bild 4. Ein Tintentröpfchen wird aus der Düse geschossen Quelle: Siemens

einzelnen Punkten zusammengesetzt werden. Besonders gut können diese Drucker Hardcopies vom Bildschirm drucken. Eingeschränkt ist die Verwendbarkeit bei Kurven- und Liniendiagrammen, da die Auflösung zu gering ist. Hier sind die **Plotter** in ihrem Element. Da der Schreiber zwischen den vom Computer berechneten Punkten nicht abgesetzt wird, interpoliert er quasi ein Kurven-Diagramm. Eine durchgezogene Linie ist die Folge.

Der Kauf eines Plotters für zu Hause lohnt sich aber nur in den wenigsten Fällen, da ein Plotter nur zum Zeichnen sinnvoll verwendet werden kann. Grafikbilder in Hochauflösung zu drucken ist praktisch unmöglich. Genauso das Schreiben von Briefen. Listings sollten sie mit einem Plotter nur dann drucken, wenn Sie über zuviel Zeit verfügen.

Ein Plotter ist kein Ersatz für einen Drucker. Vielmehr ist er als Zusatzgerät zu einem Drucker gedacht, wenn es auf saubere Zeichnungen ankommt.

Tintenstrahldrucker vereinigen die Vorteile von Nadel- und Thermodruckern. Sie sind sehr vielseitig und leise. Allerdings können keine Durchschläge wie mit dem Nadeldrucker angefertigt werden. Das Schriftbild der »Tintenspritzer« ist nicht ganz so exakt wie das gleichteurer Matrixdrucker (Bild 2).

Brauchen Sie einen schnellen Drucker fürs Büro, beispielsweise für Kontrollprotokolle, Adressenaufkleber und Rundschreiben, sind Sie mit einem leisen Tintenstrahldrucker richtig beraten. Ebenso wenn Sie zu Hause keine Familienmitglieder mit dem Lärm eines Nadeldruckers belästigen wollen.

Typenraddrucker sind nur für Textverarbeitung interessant, wenn es auf sehr schöne Schrift ankommt. Für Listingsdruck, Hardcopies und Rundschreiben sind sie so gut wie ungeeignet.

Das Problem mit dem Interface

Häufig werden wir gefragt, welches Interface das beste für einen bestimmten Drucker ist. Leider kann die Frage nicht absolut beantwortet werden. Es kommt nämlich ganz darauf an, was Sie investieren wollen. 100, 200 oder 500 Mark. Der Preis wird von Zusatzfunktionen wie C 64-Grafikzeichen-Darstellung etc. bestimmt. Bevor Sie sehr viel Geld ausgeben, beachten Sie aber, daß die Arbeit mit einem Interface um so schwieriger werden kann, je mehr Funktionen es bietet. Für die Praxis haben sich folgende Funktionen als sinnvoll erwiesen:

— CBM-ASCII-Wandlung; die Commodore-Zeichencodes werden der ASCII-Norm angepaßt (Standard).
— Groß/Grafikmodus
— Groß/Kleinmodus
— Linearkanal; hier werden die Daten ungewandelt an den Drucker geschickt. Diese Funktion ist sehr wichtig zum Drucken von Hardcopies.

Sie merken also, es gibt keinen Drucker, der alles 100prozentig macht. Vor dem Kauf sollte man sich völlig im klaren über die benötigten Funktionen sein. Funktionen die man vielleicht einmal pro Jahr braucht, sollten eine Kaufentscheidung überhaupt nicht beeinflussen. Solche Sonderfunktionen können nämlich nur den alltäglichen Gebrauch eines Druckers erschweren. In der folgenden Marktübersicht finden Sie eine Auswahl von Thermodruckern, Tintenstrahldruckern und Plottern. Daß von den letzten beiden Typen nur wenige aufgeführt sind liegt einfach am hohen Preis der Geräte. Wir haben bei 2000 Mark die Grenze gezogen.(hm)

10 Zeichen/": Normal-/ Breit-/ Fettschrift
12 Zeichen/": Normal-/ Breit-/ Fettschrift
17 Zeichen/": Normal-/ Breit-/ Fettschrift

Bild 2. Schriftprobe des Siemens PT88/89

Test: Brother EP 44



Die Brother EP 44 ist nicht nur ein Drucker, sondern auch eine voll portable Schreibmaschine. Sie wird über eine RS232-Schnittstelle angesprochen.

Die Schreibmaschine ist sehr klein (55 mm x 330 x 262 mm) und paßt mit Sicherheit in jeden Aktenkoffer. Mit ihren 2,5 kg (einschießlich Batterien) fällt sie kaum ins Gewicht.

Die Schreibmaschine hat einen Textspeicher für über 3 700 Zeichen. Dort können ganze Briefe abgelegt werden, die auch nach Ausschalten des Gerätes oder bei Batteriewechsel erhalten bleiben. Die EP 44 ist zwar auch für Netzbetrieb vorgesehen, doch fehlt das Netzgerät in der Grundausstattung.

Der Textspeicher kann auch nachträglich über das 15stellige LCD-Display bearbeitet werden. Zu den Editiermöglichkeiten gehören Löschen, Einfügen, Ergänzen und Korrigieren. Die Schreibmaschine selbst läßt kaum Wünsche offen.

Doch inwieweit läßt sich die Brother EP 44 mit einem Computer wie dem C 64 ansteuern? Der Thermodrucker besitzt dazu eine integrierte RS232-Schnittstelle. Über eine spezielle Datenleitung von Brother (CA50-CMD II) kann die EP 44 direkt an den User-Port des VC 20 oder C 64 angeschlossen werden. Über die Geräte- und Sekundäradresse 2 können nun Buchstaben, Zahlen, sowie Satz- und Rechenzeichen gedruckt werden. Grafikzeichen oder Commodore-Sonderzeichen werden ignoriert oder falsch interpretiert. Die Schreibmaschine ist deshalb nicht zum Ausdruck von Listings oder Hardcopies geeignet. Sie findet ihre Verwendung ausschließlich in der Textverarbeitung, zumal der Zeichensatz die deutschen Umlaute beinhaltet und das Schriftbild beim Druck mit Thermopapier dem einer Typenradschreibmaschine sehr nahe kommt (Bild 1). Das Schriftbild mit Plastik-Carbon-Band auf normalem Papier ist nicht ganz so gut.

Schwierigkeiten traten in der Verträglichkeit zu Textverarbeitungs-Programmen auf. Das Textverarbeitungsprogramm Vizawrite ist nicht für einen RS232-Drucker vorgesehen. Wordpro 3+ akzeptiert nicht die Geräteadresse 2. Auch mit dem Programm SM-Text treten ähnliche Probleme auf. Mit Textomat-Versionen, die nach 1983 auf den Markt kamen, soll laut Firmenauskunft das Interface zusammenarbeiten.

Easyscript, die große Ausnahme:

Nur Easyscript konnte an die EP 44 angepaßt werden. Alle anderen Textprogramme konnten hingegen auch nach längerem »Herumtüfteln« nicht in Verbindung mit der EP 44 verwendet werden.

Wichtig ist, wie der Druckerkanal eröffnet wird:

OPEN 1,2,2, CHR$(6)+CHR$(1)

Nur nach diesem OPEN-Befehl werden Texte einwandfrei ausgedruckt. Über die RS232-Leitung akzeptiert die EP 44 nur 7 Steuercodes wie:

BS (Backspace)
HT (Horizontal Tabulation)
LF (Line Feed)
CR (Carriage Return)

Wenn man schon eine programmierbare Schreibmaschine mit Textspeicher besitzt, stellt sich natürlich die Frage, ob die Programmierung auch über den Computer erfolgen kann, oder ob die gespeicherten Texte mit Hilfe des C 64/VC 20 auf Diskette oder Kassette gespeichert werden können. Leider ist das Interfacekabel hierfür nicht vorgesehen. Auf besondere Bestellung können Sie aber von Brother ein bidirektionales Interface erhalten. Der Textspeicher kann nur über die eigene Schreibmaschinentastatur bearbeitet werden. Es ist dann möglich, den in der Schreibmaschine gespeicherte Text in den C 64 zu holen. An die Brother EP 44 läßt sich direkt ein Modem anschließen. Kombiniert man sie mit einem batteriebetriebenen Akustikkoppler, so wird aus der Brother ein portables Datenterminal, mit dem man sogar aus Telefonzellen in Datenbanken oder Mailboxen herumstöbern kann.

Die Brother EP 44 ist eine wirklich gute elektrische Thermo-Matrix-Schreibmaschine mit hervorragendem Schriftbild. Wegen ihrer Größe und Gewicht eignet sie sich vor allem für kleine Arbeiten unterwegs (im Flugzeug oder Zug). Auch als Datenterminal findet die EP 44 eine gute Anwendung. Bemerkenswert ist der große Zeichensatz, der alle europäischen Sonderzeichen beinhaltet. Als Drucker für den C 64 und VC 20 eignet sich die EP 44 nur bedingt.

Für alle, die eine portable Schreibmaschine brauchen und nur ab und zu etwas über den Computer ausdrucken lassen wollen, ist die EP 44 absolut empfehlenswert.

(Christian Q. Spitzner/hm)

Info: Brother, im Rosengarten 14, Postfach 1320, 6368 Bad Vilbel, Preis 799 Mark; Druckerkabel 149,—, Netzadapter 39.—

ABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜabcdefghijklmnopqrstuvwxyzäöüß1234567890
!"§+%&/()=?,.-;:_×÷+±°²[]{}<>µ$¢£₣¥ÇçÅå¡ß;'¶#|ÆæØø¿&!@§*äàáâëèéêöòóôã

Bild 1. Der verkleinerte Zeichensatz der EP 44

Marktübersicht: Drucker

Thermodrucker

a) Anbieter b) Hersteller	Modell	Zeichenmatrix, Punkte	Grafikfähig	Auflösung Punkte/Zoll	Druckgeschw. Zeichen/s	a) Thermopapier b) Carbon-Band	Papierbreite in cm	Abmessungen in cm	a) Centronics b) RS232 c) C 64	a) Druckerkabel b) anschlußfertig an C 64 c) Mitgeliefertes Interface	Empfohlenes Interface	Preis in DM inkl. MwSt.
a) **Brother** b) Brother	HR 5 (HR 5C)	9x9	ja	k. A.	30	a, b	22	30,3x17,5x6,5	a, c	a) ja b) ja c) —	—	499,—
	EP-44 Schreib-masch.	18x24	nein	k. A.	16	a, b	22	33x26,2x5,5	b	a) nein b) nein c) Aufpreis	Brother Druckerkabel 149,—	799,—
	TC-600	18x24	nein	k. A.	16	a, b	22	34x27,6x5,8	b	a) nein b) nein c) Aufpreis	w. o.	1365,—
a) **Händler** b) Citizen	TIP680	9	k. A.	k. A.	50	a, b	22	31x21,6x7	c	a) k. A. b) ja c) ja	—	ca. 600,—
a) **Epson** b) Epson	P-40	5x9	ja	480/Zeile	40	a	11,2	20,6x12,8x4,6	a	a) nein b) nein c) nein	Centronics	448,—
a) **Star** b) Star	STX-80	5x9	ja	60	60	a	22	35,2x19x10	a	a) nein b) nein c) Aufpreis	Star-C 64-Interface, ca. 250,—	595,-
a) **Händler** b) Okidata	Okimate 20	14x14 7x14	ja	144x144	80 40	a, b, schwarz o. 3farbig	25,4	6x19x33	a, b	a, b, c in Vorbereitung	—	788,— netto
a) **Händler** b) Ricoh	TP2051C	17x9	ja, farbig	bis zu 1960 pro 8 Zoll	80	b	24,5	34,7x19,9x6,4	a, b	a) nein b) nein c) nein	Centronics	ca. 1600,— ab Herbst

Tintenstrahldrucker

a) Anbieter b) Hersteller	Modell	Zeichenmatrix, Punkte	Grafikfähig	Auflösung Punkte/Zoll	Druckgeschw. Zeichen/s	Papierbreite in cm	Abmessungen in cm	Schnittstellen a) Centronics b) RS232	a) Druckerkabel b) anschluß fertig an C 64 c) Mitgeliefertes Interface	Empfohlenes Interface	Preis in DM inkl. MwSt.
a) **Händler** b) Siemens	PT88-14N	9x9	ja	bis 102	150	22	41x31x14	a, b	a) nein b) nein c) nein	Centronics-Interface	um 2000,—
a) **Didas Computer** b) Tandberg	TTD 8802	9x9	ja	k. A.	150	22	41x31x14	a, b	a) nein b) nein c) nein	Centronics-Interface	2065,—
	TTD 8822	9x9	ja	k. A.	150	22	41x31x14	a (Epson-kompatibel)	a) nein b) nein c) nein	Centronics-Interface	2188,—

Plotter

a) Anbieter b) Hersteller	Modell	Druckgeschw. Zeichen/s	a) Flachbett b) Rolle	Papierbreite in cm	Abmessungen in cm	Schnittstellen a) Centronics b) C 64	a) Druckerkabel b) anschlußfertig an C 64 c) Mitgeliefertes Interface	Empfohlenes Interface	Preis in DM inkl. MwSt.
a) **Adcomp** b) Adcomp	X100C64	10-15	b	DIN A4	38,5x26,2x7,7	a, b (optional)	a) ja b) ja c) Softinterface	—	1980,—
	X300	5	a, DIN A3	DIN A3	57,5x44,8x10,5	a, b (optional)	a) ja b) ja c) Aufpreis (ca. 100,—)	—	2250,—
a) **Händler** b) Commodore	VC 1520	14	b	11,5	27x16x7,5	Commodore	a) ja b) ja c) ja	—	um 300,—
a) **Händler** b) Panasonic	KX-W08G Grafik-Schreibmaschine	5	b, DIN A4	DIN A4	32x26x6	k. A.	a) nein b) nein c) Aufpreis (ca. 250,—)	—	698,—
	RK-P400C	6,5	b, DIN A4	DIN A4	36,2x26,8x6,2	a, b	a) nein b) nein c) Aufpreis	Panasonic RP-K110	998,— ab August

Mit dieser Marktübersicht zeigen wir Ihnen, welche Thermo- und Tintenstrahldrucker und welche Plotter es für einen Heimcomputer wie den C 64 gibt.

In der Übersicht haben wir Drucker, die deutlich über 2000 Mark kosten, nicht berücksichtigt. Denn Geräte, die diese Preisgrenze überschreiten, sind für den Anschluß an den C 64 wenig sinnvoll, kosten sie doch fast schon viermal soviel wie der Computer selbst.

Die Marktübersicht beruht auf Herstellerangaben und erhebt keinen Anspruch auf Vollständigkeit. Außerdem sollen alle genannten Drucker an den C 64 anschließbar sein. Wenn nötig, über ein passendes Interface.

Die Vorteile der hier aufgeführten Drucker:

Thermodrucker

Diese Geräte gehören in der Regel zu den billigeren Druckern. Brauchen aber teures Spezialpapier oder extra Plastik-Carbon-Band, wenn mit normalem Papier geschrieben werden soll.

Tintenstrahldrucker

Wollen Sie einen leisen Drucker mit den Fähigkeiten eines Nadel-Matrixdruckers, sind diese Drucker die richtigen. Zum Druck brauchen Sie spezielle Tinte und saugfähiges Papier.

Plotter

Plotter sind im Prinzip keine Drucker. Vielmehr: Sie haben damit fast nichts gemeinsam. Zwar können Plotter Buchstaben schreiben, aber diese Funktion dient in der Regel nur der Beschriftung von Grafiken. Die Stärke der Plotter liegt beim Zeichnen von Grafiken, die aus Linien bestehen. Eine Linien-Grafik darf nicht mit einer Hardcopy verwechselt werden. Beim Druck von Hardcopies bleibt nämlich ein Matrixdrucker ungeschlagen.

Ein Merkmal haben alle hier vorgestellten Drucker gemeinsam: Während der Arbeit sind sie fast nicht zu hören und haben schon deshalb verschiedenste Einsatzberechtigungen. Haben Sie zu Hause kleine Kinder, die schlafen sollen, während Sie Ihrem Computerhobby nachgehen? Dann wäre ein »Tintenspritzer« ein geeigneter Ersatz für einen eventuell ins Auge gefaßten Typenrad- oder Nadeldrucker. (hm)



Mailbox für Anfänger

Das Medium Mailbox fasziniert. Der Einsteiger zeigt allerdings oft noch eine gewisse Scheu. Anhand eines Beispiels zeigen wir, wie einfach der Umgang mit DFÜ sein kann!

Als erste und wichtigste Regel gilt: Ist die Verbindung zu einer Ihnen fremden Mailbox hergestellt, sehen Sie sich die Anleitung an. Dies ist aus zwei Gründen wichtig. Zum einen werden in den meisten Fällen Menüaufrufe nur abgekürzt auf den Bildschirm gebracht, mit denen ein Neuling kaum etwas anfangen kann. In der Decates-Box wäre dies »Ihre Eingabe (A,P,D,0-6, 9)«. Erst die Anleitung gibt Aufschluß, daß nur RETURN ohne andere Eingabe das Hauptmenü ausführlich anzeigt.

Zum anderen kann es bei nicht so ausgefeilten Boxen wie Decates vorkommen, daß Fehleingaben zum Programmabsturz führen. Wenn dann der Systemoperator nicht zur Stelle ist, ist die Box erst einmal für alle Anrufer nicht erreichbar.

Ein weiteres wichtiges Thema ist der Informationsaustausch. Diese Rubrik dient hauptsächlich zwei Zwecken. Interessante Informationen möglichst vielen DFÜ-Fans bekannt zu machen und Kontakt zwischen einzelnen herzustellen. Eingaben wie »Ich grüße alle Hacker. xyz« oder »Ich war hier. xyz« machen eine Box nur unübersichtlich und treiben die Telefonkosten in die Höhe. Damit ist also niemandem gedient!

Noch ein Hinweis zum Schluß. Gerade neue Mailboxen sind oft auf die Hilfe von ihren Anrufern angewiesen. Die meisten Betreiber einer Mailbox sind für jeden Tip und Verbesserungsvorschlag dankbar. Beachtet man diese Regeln, steht einer guten Zusammenarbeit zwischen Anrufern und Mailboxbetreibern nichts mehr im Wege. (rg)

```
11:01:04 >> Sie haben 60 Min. SYSTEM-ZEIT <<

***** `ZUR ZEIT KEINE MITTEILUNGEN FUER SIE `*****

H A U P T M E N U E
-------------------
A ``A N L E I T U N G ``( Informations-System )
P ``P A R A M E T E R ``( Ausgabe-Steuerung )
D ``D I A L O G ````````( Inter-Kommunikation )
0 ``L O G O F F ````````( ENDE der Verbindung )

1 ``D E C A T E S ``````( Anschrift & Informationen )
2 ``M A I L B O X ``````( Informations-Austausch )
3 ``S O F T B O X ``````( Programm-Austausch )
4 ``T E L E B O X ``````( Mailbox-Nummern-Liste )
5 ``A N W E N D U N G ``( Programm-Ablauf )
6 ``F E R N K U R S E ``( Programmierte Unterweisung )

9 ``I N F O D A T ``````( Privates Datenbank System )

Ihre EINGABE (A,P,D,0-6,9) `==> `a

***** `A N L E I T U N G `( Informations-System ) `*****

1.) ALLGEMEINE HINWEISE

PROTOKOLL : 8 Bit, NO Parity, 1 STOP-Bit, XON, XOFF

Schliessen Sie jede Eingabe mit `> RETURN < `ab.
Ausgabe Anhalten ````( XOFF ) ```> Ctrl/S <
Ausgabe Fortsetzen ``( XON ) ````> Ctrl/Q <
Ein- Ausgabe Beenden ( MENUE ) ``> Ctrl/X < oder > ESC <
Verbindung Trennen ``( ENDE ) ```"LOGOFF" ( nicht auflegen! )


Die wichtigste Taste ist Ctrl/X oder ESC.
Mit Ihr koennen Sie Texte und Menues ueberspringen,
Ausgaben abbrechen und zur Befehlseingabe gelangen.
Somit bestimmen Sie Ihre Geschwindigkeit selbst.

Mit > RETURN < ohne Eingabe wird das Menue angezeigt.

Sie sollten uebrigens einen sinnvollen Namen beim "Logon" verwenden
da wir nur so die Moeglichkeit haben, Ihnen eine Nachricht zukommen
zu lassen !!!

H A U P T M E N U E ````11:03:00 >> Sie haben noch `58 Min. SYSTEM-ZEIT <<
-------------------

Ihre EINGABE (A,P,D,0-6,9) `==> `2

2 ``M A I L B O X `( Informations-Austausch )
---------------------------------------------

0 ``Zum HAUPTMENUE

1 `Informationen eingeben
2 `Informationen anzeigen

Ihre Eingabe `(0-2) `==> `2

===== DATE 08.05.85 =====

Msg. an ANTHONY in SAN RAFAEL, Californien, USA
VON: HEM aus D'dorf am 8.5.85
Hallo Anthony,
I was expecting your first message today as you announced
on the phone. From your brief description I learned that you
had the DECATES-LOGO on your screen. What happended that you
could not leave the said message under menue-position 2 MAILBOX?
If you have another try tomorrow please let me know bevorehand over
the phone. If there are any other problems, just give me a call!
Finally let me know the date and time of your arrival here.
Give my kindest regards to the girls and to BT (i.e. BIG THILO)
HEM


----- NEXT -----

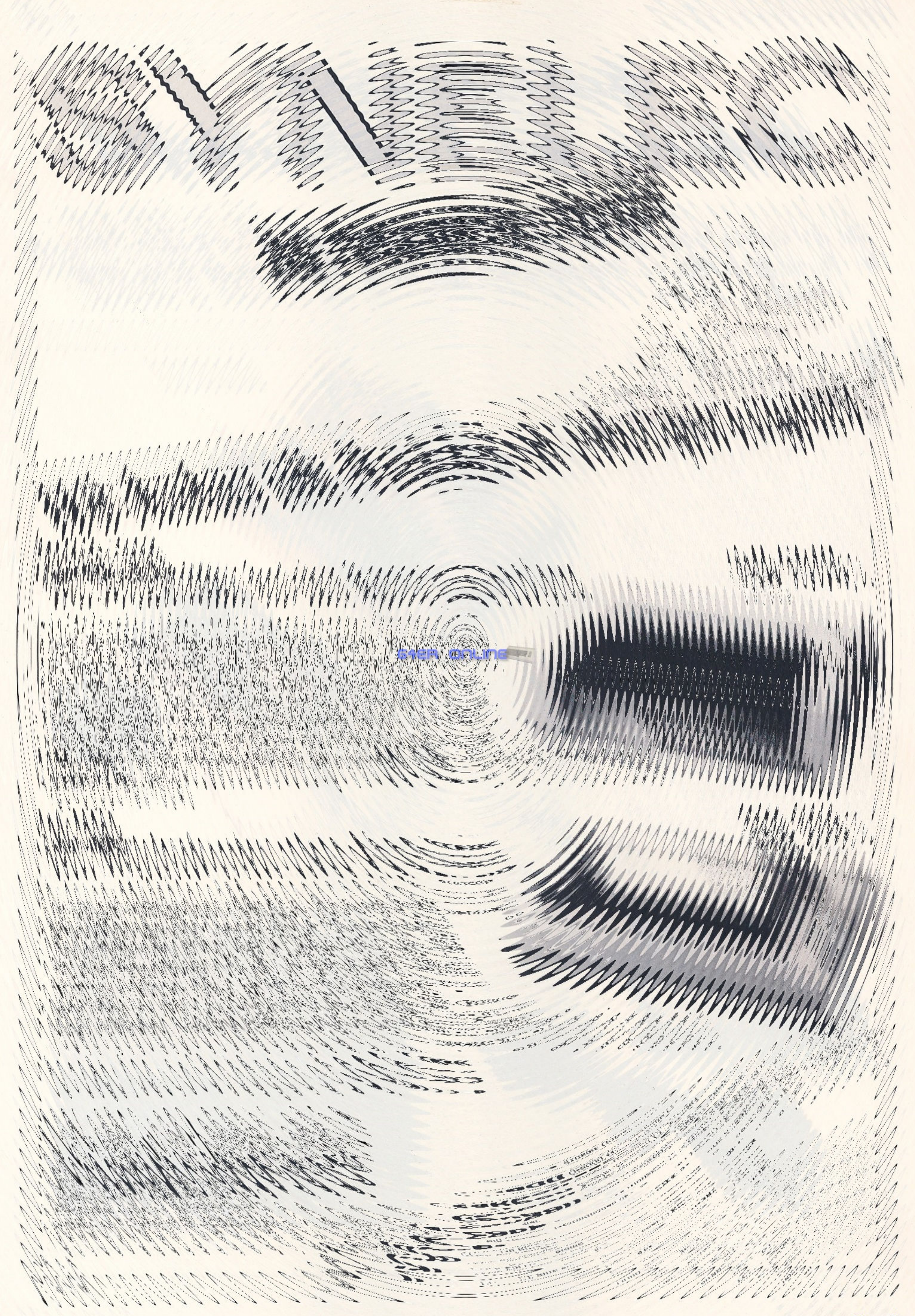
hallo phreaks !
ich wende mich an die vielen 64-besitzer, die aufihrem rechner einer
mailbox laufen haben oder laufen lassen wollen !
ich waere euch unglaublich dankbar, wenn DU, SysOp XYZ dich bei mir melden
wuerdest, da ich denke, dass a) nicht jedes rad pro mailbox einmal erfun-
den werden muss und dass b) wohl viele benutzer die ewig gleichen boxen,
die sich nur im impressum (und im namen des SysOps) unterscheiden, SysOps
wissen, dass ich z.B. von den 1001 KIS-Versionen rede, dass also viele
benutzer diese 08/15-boxen satt haben und was neues sehen wollen...
wer bock hat, mit mir zusammen, oder auch alleine, eine NEUE, INNOVATIVE
BOX auf dem c64 zu realisieren. die weder auf KIS noch auf UMC basiert,


oder eine msg an markus packenius in rmi... ok ?

uebrigens: ich betrachte meine software als freeware !
beruht weder auf KIS, noch auf UMC, noch auf MCS noch auf...

ciao
 marc aurel
```

Auszug aus Decates



Marktübersicht: Modems und Akustikkoppler

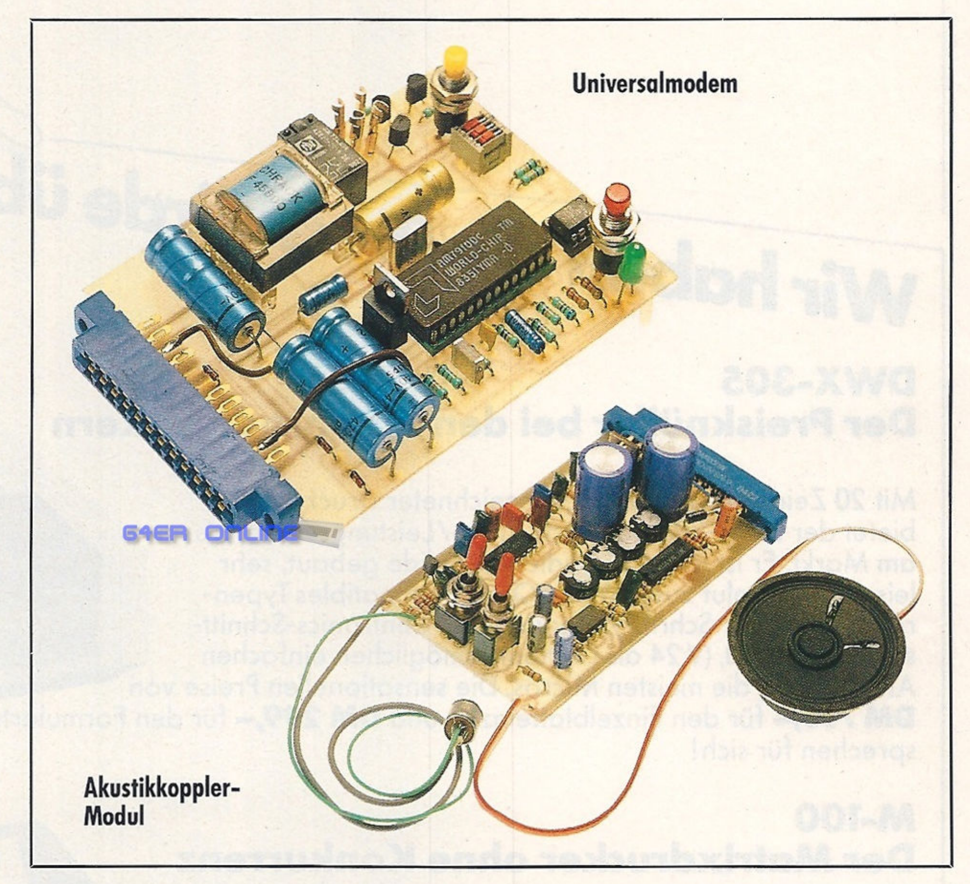
Die Preise für Modems und Akustikkoppler reichen von 138 bis 1590 Mark. Wo die Unterschiede bei den Geräten liegen, läßt sich nur im direkten Vergleich feststellen. Unsere Marktübersicht hilft Ihnen, die Übersicht zu behalten.

Die »Gerätevielfalt« bei den Modems und Akustikkopplern reicht von einfachen Platinen, an denen ein Mikrofon und ein Lautsprecher angelötet sind, bis zu optisch und technisch hochwertigen Geräten. An dieser Stelle wollen wir nur kurz auf zwei Akustikkoppler hinweisen. Alle anderen Angaben entnehmen Sie bitte der Tabelle.

Der erste Akustikkoppler ist das »Akustikkoppler-Modul« von Drust. Es handelt sich hierbei um die oben erwähnte Platine. Dieses Gerät ist nur all denen zu empfehlen, die über einen absolut schalldichten Raum verfügen. Nebengeräusche machen jede Datenfernübertragung nahezu unmöglich. Leserbriefe, die die Redaktion erreichten, bestätigten unsere Testergebnisse.

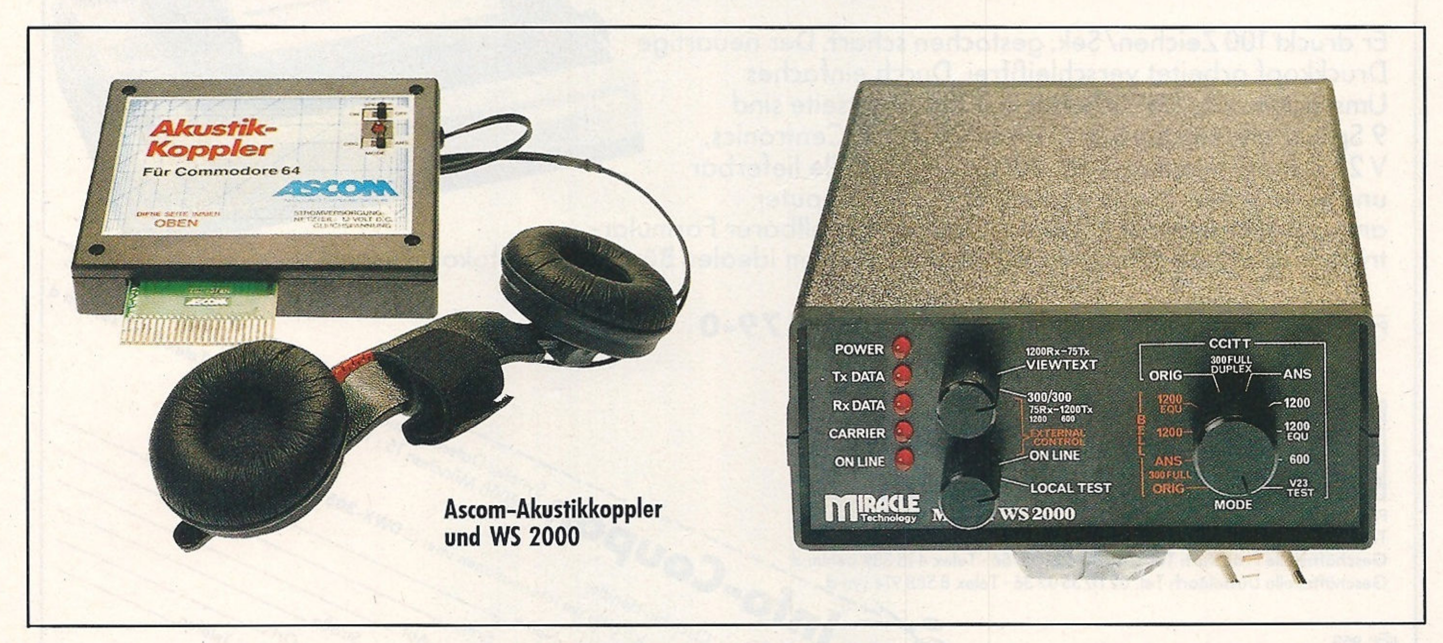
Das zweite Gerät ist der Akustikkoppler »Dataphon s 21 d« (baugleich mit dem CTK 1000). Diese Gerät überzeugt durch Übertragungsgenauigkeit und niedrigen Preis.

Unsere Marktübersicht erhebt keinen Anspruch auf Vollständigkeit und beruht auf den Angaben der Hersteller. (rg)



Universalmodem

Akustikkoppler-Modul



Ascom-Akustikkoppler und WS 2000

Anbieter/ Hersteller Gerät	a) FTZ-Nr. b) Baudrate	a) Voll-/Halb-duplex b) Originate/Answer	Akku Netzteil Batterie	a) Schnittstelle b) LEDs	a) User-Port, Expansion-Port b) Muffen-verstellbar	a) inkl. Interface/Software b) Preis (in Mark)
Stoll ST 300	a) 18.13.1996.00 b) 300	a) V b) O/A	A/N	a) V.24 b) —	a) U b) ja	a) — b) 590,—
ST 2000	a) 18.13.1909.00 b) 300 bis 1200	a) V/H b) O/A	A/N	a) V.24 b) —	a) U b) ja	a) — b) 990,—
ST 2000 (Btx)	a) 18.13.1965.00 18.13.1907.00 b) 75 bis 1200	a) H b) O	A/N	a) V.24, DBT03 b) —	a) U b) *[2]	a) — b) 990,—
HSV-Streber Dataphon s 21 d	a) 18.13.1917.00 b) 300	a) V b) O/A	A/N/B	a) V.24 b) Ans., Org., CTS	a) U b) ja	a) *[1] b) 298,—
Stockem AS-A 2470	a) keine b) 300	a) V/H b) O/A	N	a) V.24 b) Ans., Org.	a) U b) nein	a) — b) 168,—
AS-A 2480	a) beantragt b) 300	a) V b) O/A	N	a) V.24 b) Ans., Org., Ein/Aus	a) U b) nein	a) — b) 198,—
Printtechnik Universalmodem	a) keine b) 300, 1200/75	a) V/H b) O/A	—	a) — b) Online	a) U b) keine Muffen	a) I entfällt, S b) 348,— Bausatz
CTK CTK 2000b	a) 18.13.1954.00 b) 300	a) V b) O/A	A/N	a) V.24 b) DTR, CD, RTS, Ein	a) U b) *[2]	a) I b) 525,— *[3]
CTK 2000 Universalkoppler	a) 18.13.1954.00 18.13.1952.00 (Btx) b) 300, 1200, 1200/75, 75/1200	a) V/H b) O/A	N	a) V.24 b) DTR, CD, RTS, BRTS, Ein	a) U b) *[2]	a) I b) 980,— *[3]
CTK 2003 Btx	a) 18.13.1953.00 b) 75/1200	a) H b) —	A/N	a) V.24 b) DTR, CD, RTS, BRTS, Ein	a) U b) *[2]	a) I b) 980,— *[3]
CTK Minimodem 3005 S	a) 18.13.1955.00 b) 300	a) V/H b) O/A	A/N	a) V.24 b) DRS, DCD, RCV, XMIT	a) U b) ja	a) — b) 655,—
CTK 1000	a) 18.13.1917.00 b) 300	a) V a) O/A	A/N/B	a) V.24 b) Anw., Org., CTS	a) U b) ja	a) — b) 261,—
Software Express AK 300 S	a) 18.13.1897.00 b) 300	a) V b) O/A	N/B	a) V.24 b) Ein/Aus, Carrier	a) U b) ja	a) I b) 298,—
AK 300 P	a) 18.13.1897.00 b) 300	a) V/H b) A	A/N	a) V.24 b) Ein/Aus, Carrier, Received	a) U b) ja	a) I b) 625,—
AK 300	a) 18.13.1897 b) 300	a) V b) O/A	N/B	a) V.24 b) Ein/Aus, Carrier	a) U b) ja	a) I b) 385,—
K 1200/1	a) 18.13.1687.00 b) 1200	a) H b) —	A/N	a) V.24, TTL b) Ein/Aus, Carrier	a) U b) ja	a) — b) 1250,—
K&N 3001	a) beantragt b) 300	a) V b) O	N	a) V.24 b) Ein/Aus, Carrier	a) U b) ja	a) I b) 298,—
K 1200/3	a) 18.13.1687.00 b) 75/1200	a) H b) —	A/N	a) V.24, TTL b) Carrier	a) U b) ja	a) — b) 1290,—
Dynamics Ascom Akustikk.	a) 18.13.1972.00 b) 300	a) V/H b) O/H	N	a) — b) Ein/Aus	a) E b) entfällt	a) S b) 279,—
Drust Akustikk.-Modul	a) keine b) 300	a) V/H b) O/H	A/N/B	a) — b) —	a) U b) entfällt	a) S (als Listing) b) 99,—/59,— (Bausatz)
CRP WS 2000	a) keine b) 75 bis 1200	a) V/H b) O/A	N	a) V.24 b) Ein/Aus, Tx Data, Rx Data, Carr., onl.	a) U b) entfällt	a) S (als Listing) b) 798,—
Epson CX-21	a) 18.13.1808.00 b) 300	a) H b) O/A	N/A	a) V. 24 b) Ein/Aus, Carrier	a) U b) ja	a) — b) 748,—
CDI ACK 300	a) 18.13.1652.00 b) 300	a) V b) —	N/A	a) V.24 b) —	a) U b) ja	a) — b) 498,—
ACK 1200	a) 18.13.1652.00 b) 1200	a) H b) —	N/A	a) V.24 b) —	a) U b) ja	a) — b) 850,—
ACK 1200/75	a) 18.13.1652.00 b) 1200/75	a) V b) —	N/A	a) V.24 b) —	a) U b) ja	a) — b) 898,—
ACK 75/1200	a) 18.13.1652.00 b) 75/1200	a) V b) —	N/A	a) V.24 b) —	a) U b) ja	a) — b) 998,—
ACK 300 Kiel	a) 18.13.1652.00 b) 300	a) V b) —	A/N	a) V.24 b) Carrier	a) U b) ja	a) — b) 298,—
ACK U	a) 18.13.1652.00 b) 75/1200, 300	a) V/H b) —	A/N	a) V.24, DBT03 b) Carrier	a) U b) ja	a) — b) 998,—
Info-Control CPV 2000	a) 18.13.1909.00 b) 300/1200, 1200/75	a) V/H b) O/A	N	a) V.21, V.23 b) DTR, CD, RTS, BRTS, Ein	a) U b) ja	a) — b) 1214,10
Tandy AC 3	a) 18.13.1801.00 b) 300	a) V/H b) O/A	N	a) V.24 b) Carrier	a) U b) ja	a) — b) 395,—
Comko CK 211	a) 18.13.1415.00 b) 300	a) V/H b) O	N	a) V.24 b) Online, Ein/Aus	a) U b) ja	a) — b) 1370,—
CK 311	a) keine b) 300	a) V/H b) O/A	N	a) V.24 b) Online, Ein/Aus	a) U b) ja	a) — b) 1590,—

*[1] zusammen mit Interface und Software 397,—, *[2] induktive Übertragung, *[3] Mengenrabatte auch für Clubs

CPV 2000

ACK 300

Dataphon s 21 d

Minimodem 3005

Epson CX-21

Info:
Info-Control KG, Renatastr. 40, 8000 München 19

Stoll EDV-Peripherie GmbH, Lessingstr. 30, 5303 Bornheim/Köln

HSV-Streber, Pettenkoferstr. 24, 8000 München 2

Stockem Computertechnik, Berghausen 13, 5778 Meschede

Printtechnik, Stumpergasse 34, A-1060 Wien

CTK, Langenbrück 20, 5060 Bergisch Gladbach 1

Software Express Handelsg. mbH, Hugo-Viehoff-Str. 84, 4000 Düsseldorf 30

Dynamics Marketing GmbH, Große Bäckerstr. 11, 2000 Hamburg 1

Fotoelektronik Immo Drust, Landwehrstr. 5, 6100 Darmstadt

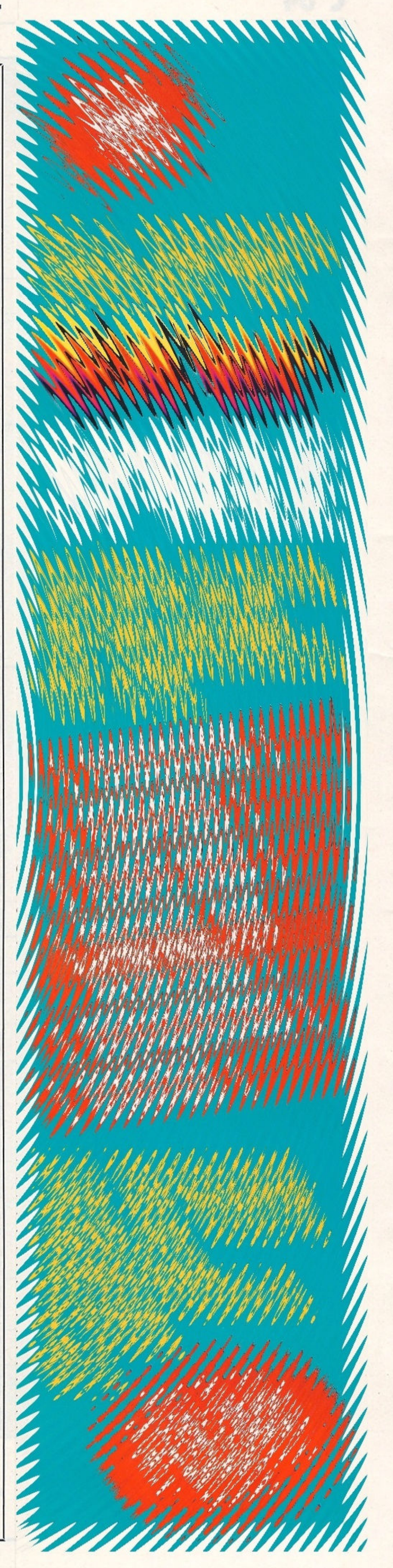
Computer Related Products, Claus F. Erbrecht, Lappenbergsallee 37, 2000 Hamburg 20

Epson Deutschland GmbH, Am Seestern 24, 4000 Düsseldorf 11

Comko Computersystemg. mbH, Marsdorfer Str. 76, 5000 Köln 40

CDI, Tauentzienstr. 1, 1000 Berlin 30

Tandy, Hermann-Lingg-Str. 30, 8000 München 2



Wie sage ich es meinem EPROM?

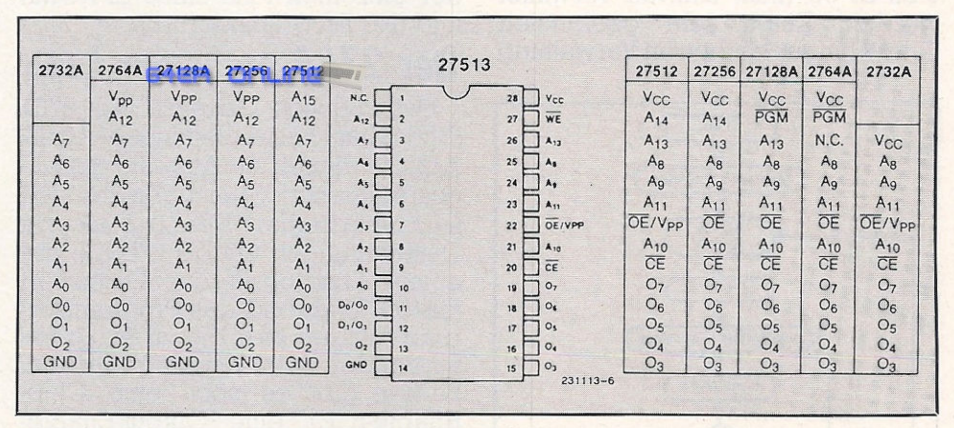
Die Programmierung von EPROMS hat ihren besonderen Reiz. Doch was ist das eigentlich, ein EPROM? Wir stellen die wichtigsten EPROM-Typen der 25er und 27er Reihe vor und zeigen Ihnen, wie man sie programmiert.

Bei Halbleiterspeichern muß man zwischen Festwert- und Schreib-/Lesespeichern unterscheiden. Die Festwertspeicher gliedern sich in ROMs (**R**ead **O**nly **M**emory), PROMs (**P**rogramable ROM), und EPROMs. Die EPROMs haben gegenüber den anderen Festwertspeichern den Vorteil, daß sie sich nach einer Programmierung wieder löschen lassen. Sie sind also für Anwendungen, in denen keine größeren Stückzahlen anfallen oder öfters Änderungen gemacht werden müssen, prädestiniert. Auch die Preise bei den Speicher-ICs haben in den letzten Jahren einen so erfreulichen Verlauf genommen, daß es sich in vielen Fällen lohnt, häufiger benötigte Programme als Modul in EPROMs zu speichern. Der Wermutstropfen bei ihrer Anwendung ist die relativ komplizierte Programmierung, für die ein eigenes Gerät benötigt wird.

EPROM-Typen

Die ersten EPROMs benötigten noch drei Versorgungsspannungen und waren nur mit großem Aufwand zu programmieren. Da ihr Preis inzwischen auch mehr als »Liebhaberpreis« zu sehen ist, soll auf diese »antiken« Stücke nicht näher eingegangen werden. Später gingen die Hersteller dazu über, Speicherbausteine mit einer 5 Volt Spannungsversorgung zu produzieren. Dadurch wurde der Aufwand, EPROMs zu programmieren, erheblich reduziert. Der erste Typ, der 2508, hatte eine Kapazität von 1 KByte. Alle EPROMs werden nach dem sogenannten Byte-wide-Prinzip gefertigt, das heißt sie haben acht Datenleitungen. Dies hat den Vorteil, daß zum Anschluß an einen 8-Bit-Mikroprozessor nur ein IC benötigt wird. Hinzu kommt, daß die Pinbelegung der EPROMs so konzipiert ist, daß sie untereinander und zu vielen statischen RAM-ICs weitgehend kompatibel sind.

EPROMs mit 2 KByte heißen 2716 oder 2516, wobei jedoch beim Kauf von 2716ern Vorsicht geboten ist, da vereinzelt auch noch ältere 3-Spannungstypen unter diesem Namen gehandelt werden. 4 KByte EPROMs gibt es im Handel unter den Bezeichnungen 2732 und 2532. Diese beiden Typen unterscheiden sich in den Pins 18 und 21. Der Grund für diese Abweichung ist darin zu suchen, daß Texas Instruments, der Hauptlieferant für 2532er, bei ihren EPROMs eine direkte Pin-Kompatibilität zu den bestehenden Masken-ROMs suchte, während die meisten anderen Hersteller die 4 KByte-EPROMs aufwärtskompatibel zu den größeren 28poligen Typen machen wollte. Das Bemühen von TI wird noch deutlicher, wenn man den 2564, den 8 KByte-Typ dieses Herstellers, betrachtet. So ist es mög-



2732A	2764A	27128A	27256	27512	27513	Pin	Pin	27513	27512	27256	27128A	2764A	2732A
	V_{pp}	V_{PP}	V_{PP}	A_{15}	N.C.	1	28	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	
	A_{12}	A_{12}	A_{12}	A_{12}	A_{12}	2	27	$\overline{WE}$	A_{14}	A_{14}	$\overline{PGM}$	$\overline{PGM}$	
A_7	A_7	A_7	A_7	A_7	A_7	3	26	A_{13}	A_{13}	A_{13}	A_{13}	N.C.	V_{CC}
A_6	A_6	A_6	A_6	A_6	A_6	4	25	A_8	A_8	A_8	A_8	A_8	A_8
A_5	A_5	A_5	A_5	A_5	A_5	5	24	A_9	A_9	A_9	A_9	A_9	A_9
A_4	A_4	A_4	A_4	A_4	A_4	6	23	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}
A_3	A_3	A_3	A_3	A_3	A_3	7	22	$\overline{OE}/V_{PP}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_2	A_2	A_2	A_2	A_2	A_2	8	21	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
A_1	A_1	A_1	A_1	A_1	A_1	9	20	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
A_0	A_0	A_0	A_0	A_0	A_0	10	19	O_7	O_7	O_7	O_7	O_7	O_7
O_0	O_0	O_0	O_0	O_0	D_0/O_0	11	18	O_6	O_6	O_6	O_6	O_6	O_6
O_1	O_1	O_1	O_1	O_1	D_1/O_1	12	17	O_5	O_5	O_5	O_5	O_5	O_5
O_2	O_2	O_2	O_2	O_2	O_2	13	16	O_4	O_4	O_4	O_4	O_4	O_4
GND	GND	GND	GND	GND	GND	14	15	O_3	O_3	O_3	O_3	O_3	O_3

231113-6

Bild 2. Der brandneue 27513 und seine kleineren Vorgänger im Vergleich (Quelle: Intel)

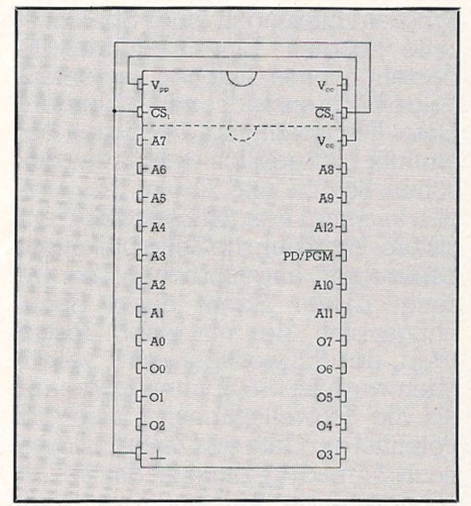


Bild 1. 2564 in einem 24poligen 8 KByte ROM-Sockel, V_{PP} muß an +5 V, CS_1 und CS_2 an Masse liegen

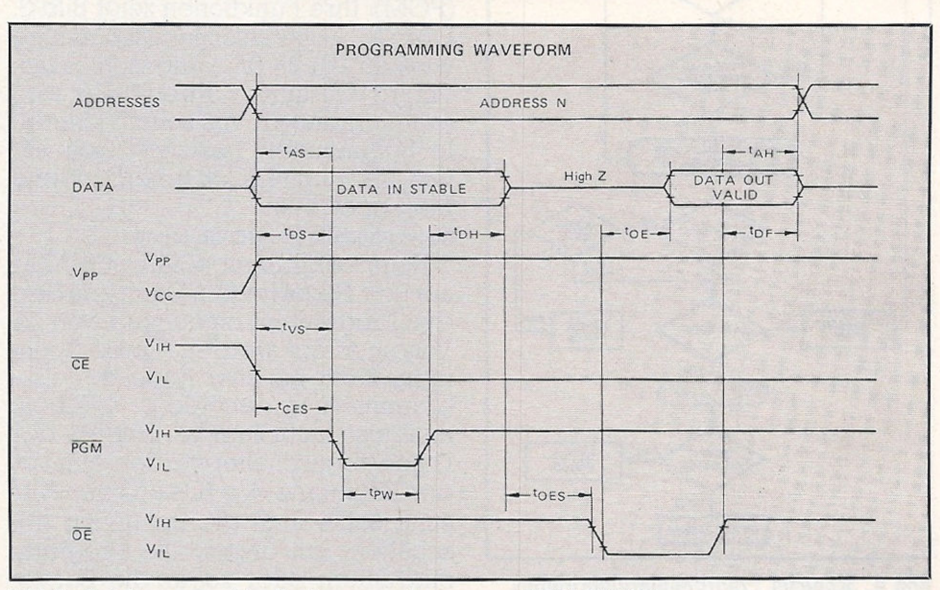


Bild 3. Zeitdiagramm für die Programmierung von 2764 EPROMs

lich, dieses 28polige IC direkt mit dem unteren Teil in einen 24poligen Sockel für ein 8 KByte-ROM zu setzen, wie zum Beispiel das Kernal-ROM des C 64, falls es gesockelt ist. Die Pins 1, 2, 27 und 28 sind einfach wie in Bild 1 gezeigt zu beschalten. Diesen Vorteil des 2564 muß man jedoch mit einem etwas höheren Preis bezahlen.

Der bei weitem verbreitetere 8 KByte-Typ ist der 2764. Er ist zur Zeit das EPROM mit dem günstigsten Preis/Bit-Verhältnis. Die Typenbezeichnung ist bei diesen EPROMs äußerst logisch aufgebaut. Die Zahl hinter der 25 beziehungsweise 27 gibt die Kapazität des Speichers in KBit an. Also: 2764 bedeutet 64 KBit oder 8 KByte. Folglich heißen die größeren Typen 27128 (16 KByte), 27256 (32 KByte) und 27512 (64 KByte). Ein EPROM mit 1 Megabit Speicherkapazität ist von NEC angekündigt, wird jedoch in einem 40poligen Gehäuse angeboten werden.

Diese immensen Speichergrößen sind aber im Moment den professionellen Anwendern vorbehalten und können von Homecomputern mit 8-Bit-CPUs nicht sinnvoll verwaltet werden. Einen sehr geeigneten Schritt hin zur leichteren Verwaltung großer EPROM-Bänke hat Intel mit ihrem 27513 (Bild 2) gemacht. Dieses EPROM ist seitenorientiert aufgebaut, das heißt es besteht aus vier EPROM-Bänken mit jeweils 16 KByte. Die Auswahl der aktiven Bank erfolgt über einen Schreibbefehl in ein internes Register des 27513. Beim Schreibbefehl werden die an den Datenleitungen D0 und D1 anliegenden Informationen übernommen und zur Seitenadresse decodiert. Beim Einschalten der Versorgungsspannung wird automatisch die Seite Null aktiviert, so daß man einen definierten Systemzustand erhält. Mit einem 27513 lassen sich also mit einem Befehl vier 16 KByte-Blöcke umschalten. Es ließen sich also auch umfangreichere Programme auf kleinere Computer umschreiben und in einem Chip unterbringen. Für den Expansion-Port des C 64 wäre dieses IC geradezu ideal. Es wird jedoch noch zirka zwei Jahre dauern, bis dieser Baustein preislich mit anderen Lösungen konkurrieren kann.

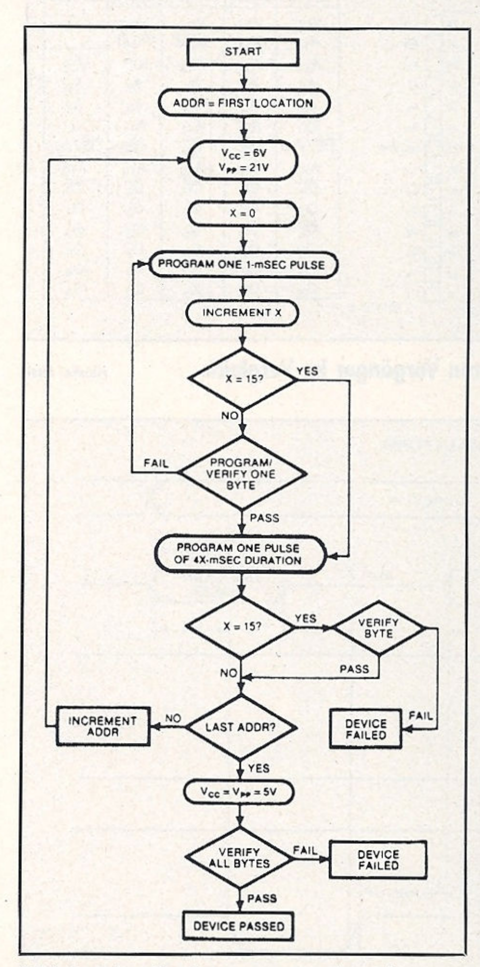


Bild 4. Schneller Programmieralgorithmus (Quelle: Intel)

Nach soviel Zukunftsmusik nun wieder zur Realität: Der Bedeutung der einzelnen Pins eines EPROMs und der Programmierung.

Der 2764 unter der Lupe

Das 28polige Gehäuse des 2764 (Bild 2) hat 13 Adreßleitungen, A0 bis A12. Sie dienen zur Auswahl der angesprochenen Speicherstelle und sind binär codiert. Der Chip hat acht Datenleitungen O0 bis O7, die bei einem Lesezugriff das Datenwort führen. Während der Programmierung müssen hier die zu programmierenden Bytes angelegt werden. Ferner gibt es noch eine Chip-Enable-(CE), eine Output-Enable-(OE) und eine Program-Leitung (PGM). Ihre Funktionen zeigt Bild 3. Die 5 Volt-Versorgungsspannung wird an Pin 28 (V_{cc}) angeschlossen, Pin 1 (V_{pp}) führt während der Programmierung 21 Volt sonst 5 Volt. Alle Spannungen beziehen sich auf das mit Pin 14 (GND) verbundene Massepotential.

Konventionelle Programmierung

Beim Programmbeginn muß PGM auf Low-Pegel liegen. Dann erst darf die Programmierspannung von 21 Volt angelegt werden. Diese Spannung kann bis zum Ende der Programmierung angelegt bleiben, muß also nicht zum Verifizieren der Daten abgeschaltet werden. Nur vor der Entnahme des ICs aus der Fassung sollte man die Spannung abschalten, um ungewollte Programmierungen oder sogar die Zerstörung des Chips zu vermeiden. Während der gesamten Programmierung muß der 2764 selektiert sein, das heißt, der Chip-Enable-Eingang auf Low-Pegel liegen. Mit OE = high werden die Ausgänge des EPROMs abgeschaltet. Jetzt werden die Adresse und das entsprechende Datenwort angelegt und, wenn diese stabil anliegen, die Programmierung durch einen 50 ms langen negativen Impuls an PGM vollzogen. Jetzt werden die Daten von den Ausgängen O0 bis O7 des EPROMs entfernt und der Inhalt der gerade programmierten Speicherzelle mit OE = Low-Pegel ausgelesen. Stimmt er nicht mit dem Sollwert überein, so ist entweder das IC oder das Programmiergerät nicht in Ordnung. Ist die Zelle korrekt »gebrannt«, kann man, nachdem OE wieder auf High-Pegel gebracht wurde, mit dem nächsten Byte fortfahren. Da die vollständige Programmierung eines 2764 mit diesem Verfahren zirka 7 Minuten dauert, haben sich die meisten Hersteller einen schnelleren Programmieralgorithmus ausgedacht (Bild 4), der die Programmierzeit für einen 2764 auf zirka 75 Sekunden verkürzt. Diese immense Zeiteinsparung ist vor allem für den kommerziellen Anwender wichtig, da sie oftmals die Anschaffung eines teuren Produktions-EPROMmers, der mehrere ICs gleichzeitig brennen kann, überflüssig macht. Aber auch für den Amateur ist eine solche Zeiteinsparung angenehm, zumal der schnelle Algorithmus keine umfangreichen Änderungen an der Hardware des Programmiergeräts benötigt.

Der schnelle Programmieralgorithmus

Um die Methode der schnellen Programmierung zu verstehen, ist es hilfreich, sich eine Vorstellung von dem zu machen, was sich bei der Programmierung in einer Speicherzelle abspielt. Eine EPROM-Zelle besteht im wesentlichen aus einem Feldeffekttransistor mit isoliertem Gate. Die Spannung des Gates beeinflußt maßgeblich den Strom zwischen Source und Drain, den beiden anderen Anschlüssen des Transistors. Wird nun die Zelle bei einem Lesezugriff angesprochen, so erzeugt dieser Strom einen Spannungsabfall, der mit einem festen Wert, der Schwellspannung, verglichen wird. Ist die Spannung kleiner als die Schwellspannung, also das Potential am Gate des Zelltransistors noch zu niedrig, dann ist der korrespondierende Ausgang des Speichers auf High-Pegel. Nach dem Löschen mit UV-Licht oder beim Kauf eines neuen Chips sind die Gates


64ER ONLINE

der Zelltransistoren ungeladen. Deshalb haben neue EPROMs immer den Inhalt »$FF«. Durch die Programmierung wird die Ladung auf dem Gate erhöht und es kann nun, wenn man während des Programmierens öfters den Speicherinhalt überprüft, der Zeitpunkt bestimmt werden, zu dem die Daten das erste Mal richtig in den Zellen stehen. Das bedeutet jedoch nicht, daß der Inhalt nun stabil für immer in den Zellen erhalten bleibt, sondern daß die Schwellspannung zum ersten Mal überschritten wurde. Die schnellen Programmieralgorithmen messen die Zeit bis dies geschieht und programmieren dann noch einmal ein Vielfaches dieser Zeiten. (Je nach Hersteller zwischen dem ein- und vierfachen).

Damit die Daten über den gesamten Temperatur- und Versorgungsspannungsbereich stabil sind, wird von allen schnellen Algorithmen eine weitere Aktion vor der eigentlichen Programmierung unternommen.

Die Versorgungsspannung wird von 5 auf 6 Volt erhöht. Die Schwellspannung, die zum Vergleich, ob eine Zelle geladen ist oder nicht, herangezogen wird, wird nämlich aus der Versorgungsspannung abgeleitet. Ist diese Schwellspannung nun etwas höher, so läßt der Algorithmus die Speicherzellen alle etwas länger brennen, als es für den normalen Betrieb eigentlich nötig wäre. Dies bringt zusätzliche Sicherheit, besonders an den Grenzen des erlaubten Betriebsbereiches und eine gute Langzeitstabilität.

Programmiergeräte

Das Komplizierteste am Selbstbau eines EPROM-Programmiergerätes ist das programmgesteuerte Umschalten von stabilen Spannungen an den verschiedenen Pins der Wechselfassung. Um ein universell verwendbares Gerät zu erhalten, sind drei Programmierspannungen (12,5 V, 21 V, 25 V) und die Möglichkeit, für den schnellen Algorithmus die Versorgungsspannung auf 6 Volt zu erhöhen, vorzusehen. Die Auswahl zwischen den EPROM-Typen kann man entweder über Schalter oder rein softwareseitig machen. Die Adressen-, Daten- und Steuerleitungen werden am einfachsten über Parallel-Ports geführt, da diese sehr leicht zu programmieren sind. Ein großes Problem bleibt die Stromversorgung. Es ist wünschenswert, ohne einen zusätzlichen Netztrafo auszukommen, nicht nur weil meistens die letzte Steckdose schon besetzt ist, sondern weil es für einen unerfahrenen Bastler die unliebsame Bekanntschaft mit 220 Volt zu vermeiden hilft. Für den C 64 bietet sich die Benutzung des User-Ports an, da er die 9 Volt Wechselspannung des Computers an den Pins 10 und 11 führt. Nachteilig für den Selbstbau ist die äußerst schwierige Programmierung der I/O-Leitungen. Als Alternative zum User-Port bliebe nur noch der Expansion-Port, der einen sehr einfachen Anschluß von Portbausteinen gestattet.

Der Nachteil dieser Alternative ist die Notwendigkeit, die Programmierspannung durch einen Gleichspannungswandler aus den 5 Volt zu gewinnen. Erstens sind solche Wandler nicht gerade billig, und zweitens wird dadurch sehr schnell die Belastungsgrenze der Computerstromversorgung erreicht. Aus den 5 Volt sollten nicht mehr als 300 mA entnommen werden, während die 9 Volt-Spannung etwas unempfindlicher auf Überlastung reagiert. Wem nach diesem kleinen Einblick in die Problematik eines solchen Projektes noch nicht der Mut zum Selbstbau eines Programmiergerätes verlassen hat, dem sei angeraten, sich die Datenblätter der Hersteller zu Gemüte zu führen. Jedoch sollte ein solcher Selbstbau wirklich dem Freak vorbehalten bleiben, da mit Spannungen gearbeitet wird, die den C 64 leicht zerstören können.

XROM und GAME

Der C 64 hat die Möglichkeit, im Expansion-Port EPROM-Module aufzunehmen, die in den Adreßbereichen $8000 bis $9FFF und $A000 bis $BFFF liegen. Um dem Computer mitzuteilen, ob in dem Expansion-Port ein EPROM steckt, hat er die Leitungen XROM und GAME. Wird XROM auf Low-Pegel gelegt, schaltet der Computer sein internes RAM von $8000 bis $9FFF ab und erzeugt für diesen Bereich einen Chip-Select (ROML), das direkt an den Chip-Select-Eingang des EPROMs gelegt werden kann. Wenn GAME auf low gelegt wird, so schaltet der C 64 seinen Basic-Interpreter ab und erzeugt für den Bereich von $A000 bis $BFFF das Chip-Select-Signal (ROMH), das ebenfalls an den Chip-Select-Eingang des EPROMs gelegt werden kann. Es ist verständlich, daß dieser Bereich nur für Maschinensprache oder andere nicht auf das Basic zugreifende Programme geeignet ist.

Autostart

Der C 64 kann Programme im $8000-Bereich als Autostart-Programme identifizieren. Als Indiz dient ihm der CBM80-Code in den Adressen von $8004 bis $8008. Zu beachten sind die nicht dem ASCII-Code entsprechenden Buchstabencodes des C 64. Sind also die Speicherzellen von $8004 an mit den HEX-Werten $C3, $C2, $CD, $38 und $30 gefüllt, so springt der Computer indirekt über die Adresse $8000 in das Programm. ($8000 = Low-Adresse und $8001 = High-Adresse). In den Adressen $8002 und $8003 steht der neue NMI-Vektor im Low-/High-Format.

Basic-Programme

Um Basic-Programme mit Autostart in EPROMs zu brennen, empfehlen wir einen Blick in die Ausgabe 9/84, Seite 163. In dem Artikel ist ein Verschiebeprogramm in Maschinensprache abgedruckt, womit ein nach $8100 verschobenes Basic-Programm wieder korrekt zurückgelegt und gestartet wird.

Mit dieser Hilfe müßte es möglich sein, ein einfaches Programm zu schreiben, das ein Basic-Programm einfach hinter den Lader schreibt und für den Prommer als HEX-Datei zum Beispiel nach $2000 schiebt.

EEPROMs

Eine neue Entwicklung der Halbleiterindustrie ist das EEPROM, ein Speicher, der sich elektrisch löschen läßt (statt wie bisher mit UV-Licht) und seinen Inhalt beim Abschalten der Versorgungsspannung nicht verliert. Diese Bausteine müssen zum Programmieren nicht aus der Schaltung entfernt werden. Die Schreibdauer liegt bei zirka 10 ms. Das ist verglichen mit RAM-ICs sehr hoch. Deshalb gibt es, vor allem von kleineren amerikanischen Firmen, Speicher, die ein statisches RAM mit einem EEPROM auf einem Chip kombinieren. Solche NOVRAMs (**N**onvolatile **R**andom **A**ccess **M**emory) haben die kurzen Zugriffszeiten eines RAMs und die Möglichkeit, mit einem Befehl den gesamten RAM-Inhalt auf einmal in die EEPROM-Zellen zu schreiben. Es müssen aber noch einige technische Probleme beseitigt werden, wie etwa die zur Zeit auf zirka 10000 Schreibzyklen pro Speicherzelle beschränkte Lebensdauer.

(Klaus Zietlow, Arnd Wängler/ah)

Frisch gebrannt ist halb gespeichert

Die andere Hälfte, die Sie zum Programmieren von EPROMs brauchen, ist ein EPROM-Programmiergerät. Unser Vergleichstest stellt einige Modelle auf eine harte Probe.

Es ist fast ein Jahr her, seit wir Ihnen das letzte Mal EPROM-Programmiergeräte vorgestellt haben. Damals galt jemand, der sich seine eigenen Speicherbausteine programmierte noch als Insider und Freak. Mittlerweile, und die täglich bei uns eingehenden Anfragen beweisen das, beschäftigt sich ein Großteil der C 64-Besitzer mit dieser Art der Datenspeicherung. Mit gutem Grund, denn erst die Programmierung von EPROMs/PROMs erlaubt es, »Ladezeiten« auf Nanosekunden zu reduzieren. So sind beispielsweise Änderungen des Betriebssystems oder neue Zeichensätze am besten im Computer aufgehoben. Der Erfolg und das Interesse der Hypra-Perfekt-Kernal-Version zeigt, wie reizvoll gerade diese Anwendungsart ist.

Um aber in der Lage zu sein, EPROMs/PROMs zu programmieren, benötigt man ein Zusatzgerät. Diese Erweiterung, die entweder am User-Port (Regelfall) oder aber am Expansion-Port (Ausnahme) angeschlossen wird, erfüllt mehrere Aufgaben. Zum einen ist es möglich, EPROMs zu lesen und deren Inhalt auf Diskette abzuspeichern, zum anderen können beliebige Programme von Diskette auf ein neues EPROM übertragen werden.

Eines unserer Testgeräte kann sogar ähnlich wie ein Floppy-Laufwerk bedient werden und erlaubt Befehle wie GET, INPUT, PRINT, LOAD, SAVE. Dadurch kann es auch die brandneuen EEPROMs (**E**lectrical **E**reasable **P**rogrammable **R**ead **O**nly **M**emorys) verarbeiten.

Keines unserer Testgeräte kostet mehr als 300 Mark. Selbst wenn man den Preis des Computers noch hinzurechnet, erhält man eine EPROM-»Brennerei«, die Leistungen wie eine professionelle Anlage hat, aber nur einen Bruchteil davon kostet.

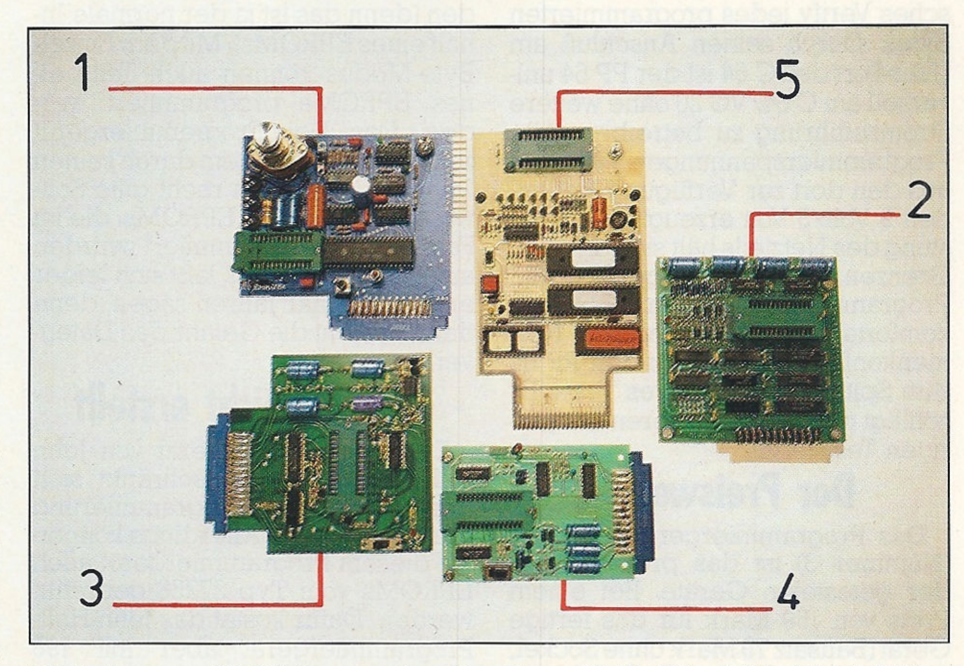


Bild 1. Gestestete EPROM-Programmiergeräte; Eprom 64 (1), Thienel (5), John-Hall (4), Dela (3), PP 64 (2)

Unser Testfeld bestand aus insgesamt sechs Geräten, von denen aber eines disqualifiziert werden mußte. Es war eine fast 100prozentige Kopie des EPROM 64 von Roßmöller (Nummer 1 im Bild). Sowohl die Software, als auch der Aufbau der Hardware wurde vom Prinzip her nicht verändert. Es wird eben nicht nur bei Rolex und Cartier geklaut, sondern auch im Zusatzgerätehandel.

Der EPROM 64 nahm nochmals am Test teil, aber als Referenzgerät. Alle Testkandidaten wurden mit den Leistungen des EPROM 64 verglichen, der ja immerhin schon eine über ein Jahr alte Entwicklung ist. Zum Test waren angetreten: Das Thienel-Programmiergerät (Nummer 5), das Gerät von John Hall (Nummer 4), das Dela-Programmiergerät (Nummer 3) und der PP 64 von Indutronic (Nummer 2).

Der Super-EPROMmer

Gleich vom ersten Moment an hat der PP 64 von Indutronic (Nummer 2) einen sehr guten Eindruck gemacht. Er ist sehr sauber aufgebaut, bedarf keiner zusätzlichen Schalter mehr auf der Platine, hat ein ausführliches Handbuch und besticht durch seine Leistungsfähigkeit. Mit dem PP 64 können nicht nur so gut wie alle EPROM-Typen (siehe Tabelle) programmiert werden, er verfügt auch über eine Funktion, mit der EPROMs von Basic aus wie ein Floppy-Laufwerk angesprochen werden. Man kann sich sogar eine Art Directory des EPROMs anlegen. Auch Befehle wie OPEN und PRINT # sind möglich, die Geräteadresse des Gerätes ist in diesen Fällen 16. Zusätzlich zu dieser Funktion bietet die mitgelieferte Software einiges an Bedienungskomfort, der eigentlich kaum Wünsche offen läßt (Tabelle).

Der PP 64 beweist, daß nicht unbedingt jedes EPROM mit ein und demselben Programmieralgorithmus gebrannt werden muß. Je nach gewähltem EPROM-Typ wird einer von vier verschiedenen Algorithmen verwendet. Das stellt nicht nur sicher, daß EPROMs die für sie optimale Programmierzeit bekommen, sondern auch eine lange Lebensdauer der EPROMs. Bei einem immer noch relativ hohen Preis für EPROMs ist das besonders wichtig. Es ist ein einfaches Rechenexempel: Ein schlechtes Programmiergerät zerstört so manches EPROM schon nach kurzer Zeit. Ein gutes Programmiergerät erlaubt aber die EPROMs in ihrer ursprünglichen Funktion, nämlich als mehrfach lösch- und beschreibbaren Speicherbaustein zu verwenden. Die vier Programmieralgorithmen des PP 64 garantieren das. Allerdings ist er mit einem Preis von 1953 österreichischen Schillingen (279 Mark)

auch nicht gerade billig. Zu den weiteren Besonderheiten gehören eine Löschroutine beim Programmieren von EEPROMs und ein automatisches Verify jedes programmierten Bytes. Durch seinen Anschluß am User-Port des C 64 ist der PP 64 universell am C 64/VC 20 ohne weitere Stromzuführung zu betreiben. Alle Programmierspannungen werden aus den dort zur Verfügung stehenden 9 Volt/5 Volt erzeugt. Die Belastung des Netzteils hält sich dabei in Grenzen. Seine weit gefächerten Programmierfähigkeiten, seine komfortable Software und sein Normenkonformität machen den PP 64 zum Spitzenreiter unseres Tests. Er schlägt damit auch unseren letztjährigen Testsieger.

Der Preiswerte

Das Programmiergerät von Dela (Nummer 3) ist das preiswerteste der getesteten Geräte. Bei einem Preis von 159 Mark für das fertige Gerät (Bausatz 79 Mark ohne Sockel, 109 Mark mit Textool-Sockel) bietet das Dela-Programmiergerät durchaus sehenswerte Leistungen. Zwar programmiert es nur EPROMs vom Typ 27xxx bis hin zum 27128, da diese aber wohl die gebräuchlichsten sein dürften, reicht es in den meisten Fällen. Auch das Dela-Programmiergerät wird am User-Port angeschlossen und kommt ebenfalls ohne zusätzliche Schalter aus. So gesehen ist er sogar besser als unser Referenzgerät, bei dem ja immer noch zwei Schalter bedient werden müssen. Die zugehörige Software erfüllt fast alle Ansprüche. Es ist sogar ein Monitor eingebaut, dessen Funktionen durchaus professionell sind. Er kennt die Befehle LOAD, SAVE, FILL, DISASSEMBLE, MEMORYDISPLAY, HUNT, TRANSFER, GO, WALK sowie eine Hex-Dez-Umrechnung. Nur zwei Dinge sind es, die beim Arbeiten mit dem Dela-Programmiergerät fehlen: Zum einen ist keine Funktion vorgesehen, um den Speicherinhalt des EPROMs mit dem des C 64 zu vergleichen, zum anderen wird keine Plausibilitätskontrolle der Eingaben durchgeführt. Eingabefehler beim Programmieren sind deshalb nicht auszuschließen.

Die wesentlichste Besonderheit des Dela-Programmiergerätes sind seine zwei Programmiergeschwindigkeiten. Im Normalmodus wird mit 50 Millisekunden programmiert, ein 2764-EPROM benötigt auf diese Weise ungefähr 7 Minuten. Im Fast-Modus wird die Programmierzeit auf 2 bis 5 Millisekunden pro Byte herabgesetzt. Das gleiche EPROM ist dann schon nach 22 Sekunden fertig. Positiv ist auch, daß leere Bytes ($FF) nicht programmiert werden (denn das ist ja der normale Inhalt eines EPROMs). Mit dem Einzel-Byte-Modus können auch Teile eines EPROMs programmiert werden. Das Dela-Programmiergerät zeichnet sich vor allem durch seinen Preis und die doch recht gute Software aus. Wie sich EPROMs, die im Fast-Modus programmiert wurden auf Dauer verhalten, läßt sich leider erst in ein paar Jahren sagen, denn dann besteht die Gefahr des Datenverlustes.

Module leicht erstellt

Das Programmiergerät von John Hall (Nummer 4) beschränkt sich ebenfalls auf die Programmierung der 27xxx-Reihe. Allerdings können mit diesem Programmiergerät auch EPROMs vom Typ 27256 gebrannt werden. Dafür kostet das John-Hall-Programmiergerät aber mit 198 Mark auch 30 Mark mehr als das Dela-Programmiergerät. Damit sind die besonderen Fähigkeiten des John-Hall-Programmiergerätes allerdings bei weitem noch nicht erschöpft. Zwar muß bei ihm immer noch ein Schalter betätigt werden, der Rest wird aber von der sehr komfortablen Software eingestellt. Sie erlaubt vor allem das freie Verlegen der Treibersoftware in verschiedene Bereiche, so daß es in den seltensten Fällen zu Konflikten mit Programmen, die einen bestimmten Speicherbereich belegen, kommt. Die eigentliche Besonderheit dieses Gerätes ist aber die Fähigkeit, von jedem Basic-Programm bis zu 8 KByte Länge ein selbststartendes Modul zu erstellen. In diesem Modul werden Verschieberoutinen und Initialisierungsbefehle automatisch mitprogrammiert. Basic-Programme mit mehr als 8 KByte müssen aus mehreren Programmiervorgängen zusammengesetzt werden.

Das John-Hall-Programmiergerät verfügt über einen automatischen Leertest und ist voll menügesteuert. Zusammen mit der ebenfalls von diesem Hersteller erhältlichen 128-KByte-Platine können somit regelrechte ROM-Erweiterungen von enormer Größe gebaut werden. Da das John-Hall-Programmiergerät nur einen Programmieralgorithmus von 50 Millisekunden besitzt, dauert das Programmieren eines 2764-EPROMs zirka 7 Minuten. Gemessen an seinen Fähigkeiten, kann diesem Gerät ein befriedigendes Preis-/Leistungs-Verhältnis bescheinigt werden.

Es geht auch am Expansion-Port

Das Thienel-Programmiergerät wird im Gegensatz zu den anderen Geräten am Expansion-Port angeschlossen. Er benötigt deshalb auch eine zusätzliche Stromversorgung. Dafür zeichnet sich das Gerät aber durch seine enorme Leistungsfähigkeit aus. Die Treibersoftware befindet sich direkt auf dem Gerät und startet automatisch (abschaltbar). Zusätzlich ist sogar noch ein weiterer Steckplatz für ein 2764-EPROM vorgesehen. Bei diesem Gerät müssen keinerlei Schalter bedient werden, denn die Auswahl der EPROM-Typen wird rein über Software gesteuert. Fast alle derzeit wichtigen Typen sind programmierbar.

Fortsetzung auf Seite 160

Merkmal	Gerät	EPROM 64 Roßmöller	Dela	PP 64 Indutronic	Thienel	John Hall
Anschluß		User-Port	User-Port	User-Port	Expansion-Port	User-Port
Auswahl des Eprom-Typs über		Schalter	Software	Software	Software	Schalter
Eprom-Typen:						
2716—27128		x	x	x	x	x
27256		x	—	x	x	x
2532/2564		x	—	x	x	—
2508/2708		x	—	—	—	—
2516/2716		x	x	x	—	—
2564		x	—	x	x	—
2758/2815/2816/8016/ 68764/68766/5133/5143/ 52B13/X2804A/X2816A		—	—	x	—	—
Sockel Textool		x	x	x	x	x
Basic-Programm brennen		—	—	x	—	x
Preis		198 Mark	159 Mark	279 Mark	ab 248 Mark	198 Mark
Bezugsquelle		Roßmöller GmbH, Finkenweg 1, 5309 Meckenheim, Tel. 02225/14488	Dela Elektronik, Krefelder Str. 66, 5000 Köln, Tel. 0221/725336	Merlin Data Vertrieb, Römerfeld 12, 8261 Tittmoning, Tel. 09683/1671	Stephan Thienel, Sandg. 22, 8602 Burgebrach, Tel. 09546/1456	John Hall, Spaltingstr. 1, 2000 Hamburg 1, Tel. 040/234262

Tabelle der fünf Testkandidaten

Aus eins mach vier

Wir stellten schon mehrere geänderte Betriebssysteme für den C 64 vor. Das beste Beispiel ist das Hypra-Perfekt. Diese Bauanleitung setzt der ganzen Sache die Krone auf: Absturzfreies Umschalten zwischen vier Betriebssystemen.

Die Idee zu dieser Schaltung lieferte eigentlich das Hypra-Load-Programm in der Betriebssystemversion, welches ein neues EPROM im Kernalsteckplatz U 4 erfordert. Mit etwa 98 Prozent der Programme funktioniert dies hervorragend. Für die restlichen 2 Prozent, mit denen es Schwierigkeiten gibt, hilft allerdings nur, das Originalkernal wieder einzusetzen. Meistens ist das der Fall, wenn Kassetten-Routinen benötigt werden. Eine Umgehung des ständigen Ein- und Ausbaus bietet sich mit einer Betriebssystemumschaltung an. So könnte ein Programm mit Hypra-Load geladen und anschließend auf das Originalsystem zurückgeschaltet werden. Dadurch wäre eine 100prozentige Programmverträglichkeit gewährleistet. Allerdings nur, wenn beim Umschalten, der Computer nicht »abstürzt«.

Aus diesen Gedanken heraus wurde diese Schaltung entwickelt. Sie hat die oben beschriebenen Vorteile und bietet noch die Möglichkeit der Umschaltung zwischen dem Original-Kernal und drei zusätzlichen Betriebssystemen.

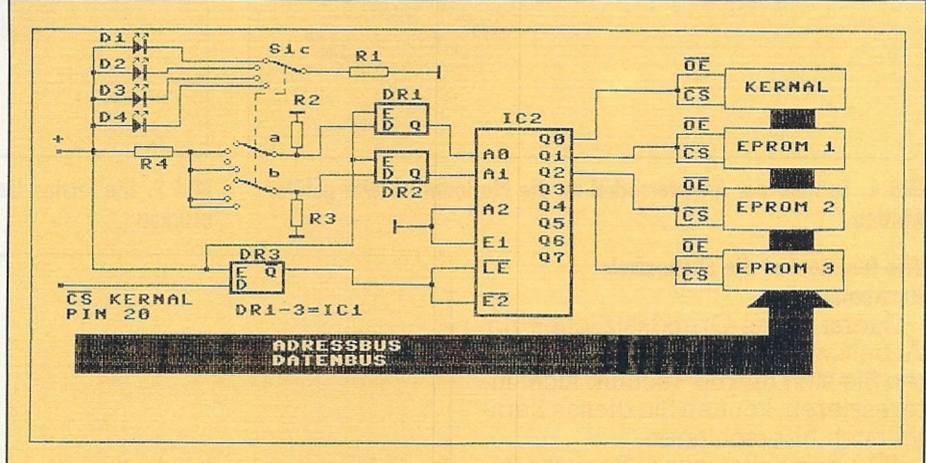


Bild 1. Schaltbild der Umschaltplatine. Es wurde übrigens mit dem Malprogramm »Hi Eddi« aus der Ausgabe 1/85 64'er gezeichnet.

Etwas Theorie

Wie wird dem Computer ein Umschalten des Betriebssystems mitgeteilt? Für den Computer stellt das Betriebssystem eine Ansammlung von Befehlen, Sprungadressen und Routinen in einem bestimmten Adreßbereich dar. Beim C 64 ist es der Bereich 57344 bis 65535 ($E000 bis $FFFF). Will der Computer aus irgendeinem Grund auf das Betriebssystem zugreifen, aktiviert er den Speicherbaustein für diesen Adreßbereich und legt ihn auf den Adreß- und Datenbus. Was für eine Art Speicherbaustein (ROM, RAM oder EPROM) das ist, ist ihm völlig egal. Hauptsache er findet in dem Adreßbereich einen Speicher vor.

Für uns ist es aber nun sehr wichtig zu wissen, nach welchem System der Computer seine Speicherbausteine verwaltet, damit ohne Absturz ein neues Betriebssystem aktiviert werden kann. Das Zusammenspiel der einzelnen RAM/ROM-Bausteine wird von einem einzigen IC gesteuert. Dieses IC befindet sich im Steckplatz U 17 und ist ein sogenannter PLA-Chip (Programable-Logic-Array). Dieses IC wird bei der Fertigung nach Herstellerangaben programmiert und erhält so besondere Logikeigenschaften, die eine Vielzahl von herkömmlichen Gatterfunktionen nachbilden. Dieser PLA-Baustein ist im C 64 dem Prozessor untergeordnet und nimmt diesem weitgehend die Verwaltung der Speicherbausteine im gesamten Adressierungsbereich ab. Die PLA, auch Adreßraummanager genannt, sorgt dafür, daß auf entsprechende Befehle der CPU immer der richtige Baustein angesprochen wird und daß immer nur ein RAM/ROM-Baustein aktiv auf dem Adreß- oder Datenbus zugreift, damit keine »Kurzschlüsse« entstehen. Jeder RAM/ROM-Baustein ist über eine Leitung mit dem Manager verbunden. Sie heißt Chipselect- oder Enableleitung. Sie aktiviert die einzelnen Speicher über ein Low-Signal und blendet dadurch den entsprechenden Adreßbereich ein. Um beispielsweise das Betriebssystem zu aktivieren, legt der Manager die Kernalleitung auf Low-Pegel (Pin 20 am Kernal-ROM) und hält alle anderen Leitungen auf High-Pegel. Damit ein anderes Betriebssystem aktiviert wird, muß man nur die Kernalleitung zum anderen Betriebssystem durchschalten und es natürlich auch mit den Adreß- und Datenleitungen verbinden. Will man drei neue und das alte System benutzen, benötigt man eigentlich nur einen 4fach Umschalter, der die Chipselect-Leitung verteilt. Allerdings hat die ganze Sache einen Haken.

Der Computer läuft mit einer Taktfrequenz von 1 MHz. Entsprechend schnell werden die einzelnen Speicher-ICs auch geschaltet. Würde man mit einem einfachen Umschalter ein anderes Betriebssystem auswählen, wäre während der Umschaltphase, die ja immerhin selbst bei »Fingerakrobaten« einige Millisekunden dauert, gar kein Betriebssystem verfügbar und der Computer würde ins »Leere« springen. Um das zu verhindern, darf eigentlich nur umgeschaltet werden, wenn auf das Betriebssystem gerade nicht zugegriffen wird. Hier setzt diese Schaltung an. Sie speichert den Umschaltimpuls so lange, bis das Kernal-ROM nicht mehr aktiv ist und gibt dann das andere angewählte Betriebssystem frei.

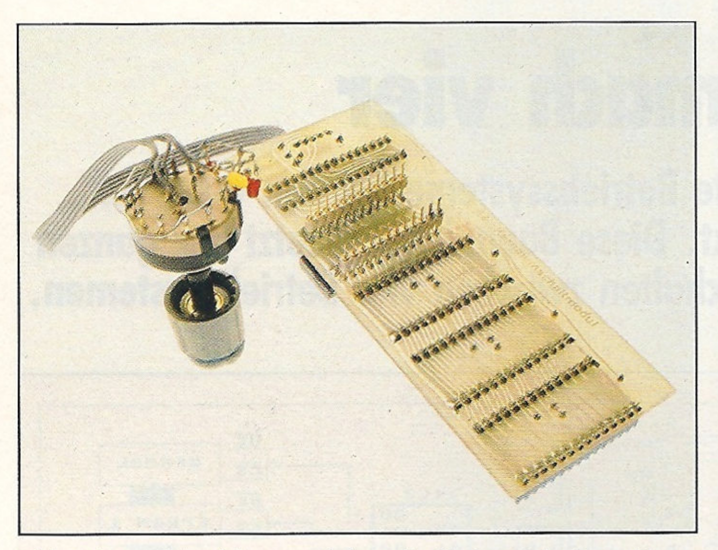


Bild 4. So muß der Adaptersockel an die Platinenrückseite gelötet werden.

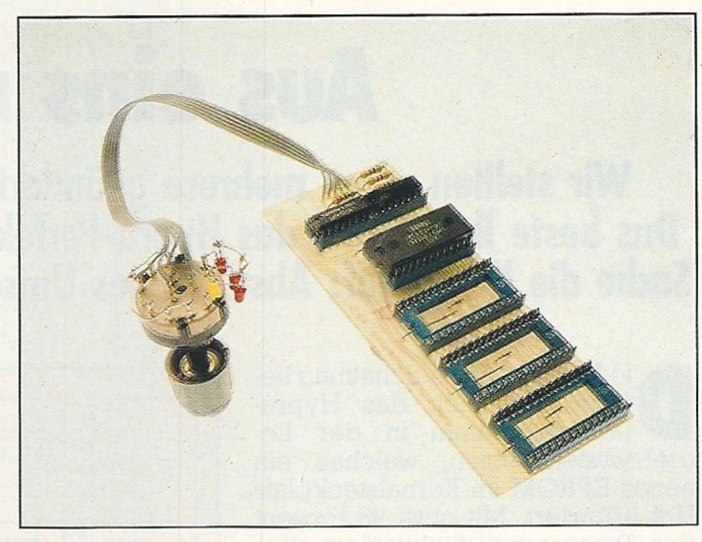


Bild 5. Die fertige Umschaltplatine. Achten Sie auf die Drahtbrücken

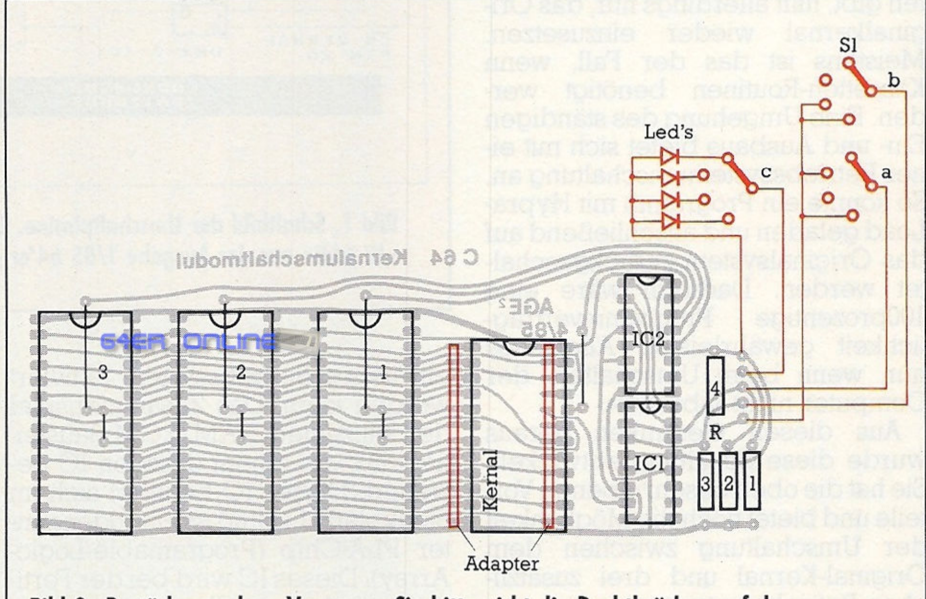


Bild 3. Bestückungsplan. Vergessen Sie bitte nicht die Drahtbrücken auf der Bestückungsseite einzulöten.

Wie funktioniert die absturzfreie Umschaltung?

Zuerst etwas Grundsätzliches zur Arbeitsweise der beiden ICs. Sollten Sie sich für die Technik nicht interessieren, können Sie dieses Kapitel auch überschlagen.

IC 1 beinhaltet vier D-Register (bistabile Gatter, Flip-Flops) mit zwei Eingängen E und D (Bild 1) und zwei Ausgängen Q und $\overline{Q}$. Der Eingang E ist ein Enable, das heißt ein Freigabe-Eingang. Liegt dieser Eingang auf High-Pegel, wird das an D anliegende Signal sofort zum Ausgang Q übertragen. $\overline{Q}$ führt dagegen immer das entgegengesetzte Signal zu Q. Deshalb auch der Strich über dem Buchstaben. Wird an E ein Low-Signal gelegt, wird das in diesem Augenblick an D anliegende Signal gespeichert. Das heißt, daß Signaländerungen an D nicht wirksam sind, solange E auf Low-Pegel liegt. Das D-Register ist also nichts anderes als ein 1-Bit-Speicher, den man freigeben oder sperren kann. DR 1-3 sind völlig identisch.

Die technische Bezeichnung für IC 2 lautet: 3-Bit-Binärdecoder/Demultiplexer (3-8) mit Adressen-Zwischenspeicher. Das hört sich komplizierter an als es ist. Die drei Eingänge A0 bis A2 (Bild 1) wirken so, wie die D-Register DR1-3. Der Eingang LE (Latch Enable) bestimmt die Funktion der Adreßeingänge A0 bis A2. Liegt LE auf High-Pegel, werden die zuletzt an den Eingängen A0 bis A2 anliegenden Pegel gespeichert. Liegt dagegen ein Low-Pegel an LE, werden die Signale an A0 bis A2 an die Ausgänge weitergegeben. Nun sind bei diesem IC die Ausgänge der Speicher nicht herausgeführt, wie bei IC 1, sondern führen auf einen internen 1 aus 8-Decoder. Die Ausgänge Q0 bis Q 7 sind die Ausgänge dieses Decoders, der wie folgt arbeitet: Von seinen acht Ausgängen kann immer nur einer Low-Pegel führen. Welcher, das bestimmt die als 3-Bit-Binärzahl an den Eingängen A0 bis A2. Mit 3 Bit lassen sich ja bekanntlich die Dezimalzahlen von 0 bis 7 darstellen. Als Beispiel: an A0 bis A2 liegt die Bitkombination 101 an. Dezimal ist das 5, somit liegt Q5 auf Low-Pegel, und zwar so lange, wie diese Bitkombination anliegt. Alle anderen Ausgänge führen High-Pegel.

Weiterhin hat IC2 zwei Freigabe-Eingänge E1 und $\overline{E2}$, die vollkommen unabhängig von den anderen Eingängen arbeiten. Mit E1 oder $\overline{E2}$ können alle Ausgänge Q0-Q7 gleichzeitig auf High-Pegel gelegt werden. Dazu muß an E1 ein High- oder an $\overline{E2}$ ein Low-Signal anliegen.

Anschaulicher wird es an einem Beispiel: Wir wollen mit Hilfe eines 3-Bit-Wortes irgendein Signal, sagen wir ein Rechtecksignal, nacheinander an vier verschiedene Stellen leiten. Dazu legen wir das 3-Bit-Wort an die Adreßeingänge A0 bis A2 an. E1 wird auf Masse gelegt, da nur ein Freigabeeingang benötigt wird. Wie bringen wir aber nun zum Beispiel Ausgang Q5 dazu, unser Rechtecksignal zu übertragen? Zuerst wird an A0 bis A2 die Bitkombination 101 angelegt, mit der wir Ausgang Q5 bestimmen. Dann wird das Rechtecksignal gleichzeitig an LE und an $\overline{E2}$ gelegt. Liegt das Rechtecksignal gerade auf High-Pegel, sind alle Ausgänge, somit auch Q5, auf High-Pegel. Geht das Rechtecksignal auf Low-Pegel, gibt $\overline{E2}$ die Ausgänge frei, und das Low-Signal

an LE überträgt sich auf den adressierten Ausgang; hier Q5. Geht das Rechtecksignal wieder auf High-Pegel, beginnt alles von vorn. Das ist genau das, was wir wollen. Wird an A0 bis A2 jetzt ein anderer Ausgang adressiert, wird das Eingangssignal auf den anderen übertragen.

Nachdem nun die Funktion der einzelnen ICs etwas verdeutlicht wurde, zur eigentlichen Umschaltung. Die Auswahl des Betriebssystems erfolgt über die Schalter S1 (a und b). Die Auswahlsignale gelangen nicht direkt auf die Adreßeingänge, sondern erst über die D-Register DR1 und DR2. Die beiden Register haben die Aufgabe, den Pegelwechsel vom Schalter »schön rechteckig« zu machen.

Hinweise zum Aufbau

Das Layout für die Umschaltplatine zeigt Bild 2, die Stückliste Tabelle 1. Beachten Sie beim Ätzen, daß einige Leiterzüge sehr schmal sind und nicht verschwinden.

Nach erfolgreicher Herstellung der Leiterplatte wird diese gebohrt. Die angedeuteten Löcher zum Anschluß der Steckverbinder dürfen nicht durchbrochen werden, da die Steckverbinder als Adaptersockel später direkt von hinten auf die Platine gelötet werden. Zuerst sollten die Drahtbrücken eingesetzt werden und dann die Widerstände und IC-Sockel. Zur Befestigung der Adaptersockel lötet man diese erst an den zwei Endpunkten an und richtet sie rechtwinklig aus. Danach sollte man ausprobieren, ob die ganze Sache auch paßt. Erst dann sollten die Steckverbinder endgültig mit der Platine verlötet werden. Den 4fach-Umschalter schließt man am günstigsten mit Flachbandleitung an. Um Leiterzüge auf der Platine zu sparen, wird die eigentliche Verdrahtung des Schalters erst an den Anschlußpins des Schalters selbst durchgeführt. Der Schalter wird laut Schaltplan »binär« verdrahtet: die Schalterausgänge müssen beim Weiterschalten die Binärzahlen 00-01-10-11 liefern. Das ist einfach möglich, indem man die entsprechenden Schaltereingänge mit R4 (5 Volt) verbindet oder offen läßt. Durch das Offenlassen entsteht über R2 oder R3 Low-Pegel.

Die Leuchtdioden sollen eigentlich nur dem Spieltrieb von Computerfreaks entgegenkommen. Die LEDs werden über die noch freien vier Kontakte des Umschalters geschaltet. Die Versorgungsleitung für die Leuchtdioden kommt über R1 von der Platine. Wollen Sie die LEDs nicht einbauen, kann der 470-Ohm-Widerstand selbstverständlich ersatzlos entfallen. Es ist dann auch nur noch ein 2x4-Umschalter notwendig.

Ist die Platine nun so weit bestückt, können die Betriebssysteme eingesetzt werden. Dazu kommt das Kernal-ROM aus Steckplatz U4 in die 24polige Fassung. Wenn Sie eine C 64-Version mit fest eingelötetem ROM haben, sollten Sie sich von einem Lötkolben-Profi eine Fassung einbauen lassen. In die Fassungen 1 bis 3 auf der Platine werden die Betriebssystems-EPROMs gesteckt. Achten Sie unbedingt darauf, die EPROMs und das ROM richtig herum in die Fassungen zu stecken (Bestückungsplan, Bild 3). Möglich sind nur 2764- oder 27C64-Typen, die ohne Schwierigkeiten zu beschaffen sind. Die C-Typen sind CMOS-Ausführungen und bieten den Vorteil der geringen Stromaufnahme. Hat man nur ein neues Betriebssystem, muß dieses unbedingt in Fassung 1 eingesteckt werden, das nächste in Fassung 2 und das dritte in Fassung 3. Sollte eine Fassung frei bleiben, darf diese natürlich nicht über den Schalter angewählt werden. Ein Absturz des Computers bliebe mit Sicherheit nicht aus.

Ist alles soweit fertig (vorher nochmal auf Lötbrücken untersuchen!), kann die Platine in den Steckplatz U4 eingesetzt werden. Die EPROMs müssen nach dem Einsetzen links von U4 stehen und alle Markierungskerben der ICs auf die C 64-Rückseite zeigen. Der Wahlschalter kommt zunächst in Stellung 1. Nach dem Einschalten des Computers sollte dieser sich normal melden. Dann kann umgeschaltet werden. Schaltet man auf Hypra-Perfekt, dürfte auf dem Bildschirm keine Veränderung feststellbar sein. Aber an der geänderten Funktionstastenbelegung wird es schnell klar, daß man ein anderes Betriebssystem aktiviert hat. Noch ein Wort zur Umschaltung: Der Computer wird natürlich nur dann nicht »abstürzen«, wenn innerhalb der Betriebssysteme zur Umschaltzeit die gleichen Routinen benutzt werden. Konkret: Benutzt man ein System mit Hypra-Load, darf natürlich während des Ladens nicht auf das normale Kernal umgeschaltet werden, weil die Laderoutine mit der originalen nicht identisch ist. Bei anderen Betriebssystemen muß man das von Fall zu Fall ausprobieren. Am sichersten ist ein Umschalten immer im Direktmodus. Kaputtgehen kann während des Umschaltens nichts. Spätestens nach einem Reset ist der Rechner wieder einsatzbereit.

(Andreas Gerzen/hm)

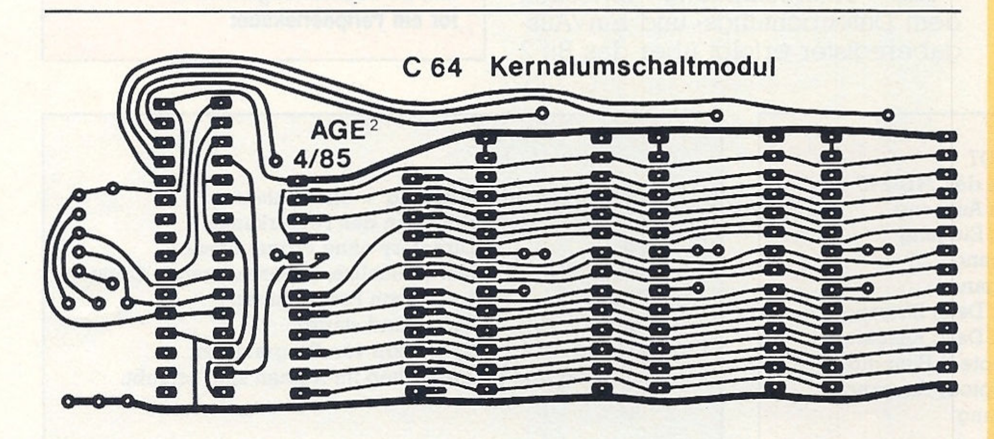


Bild 2. Das 1 : 1-Platinenlayout der Umschaltplatine.

IC 1	74 LS 75
IC 2	74 LS 137
D1,D2,D3,D4	LED 3 mm
R1	470 Ω
R2,R3	4,7 kΩ
R4	330 Ω
S1	Umschalter 3x4 trennend
IC-Fassungen	
16polig	2 Stück
24polig	1 Stück
28polig	3 Stück

Schaltdraht für Brücken
5fach-Flachbandleitung, etwa 20 cm
Adaptersockel
Da sie schwer zu beschaffen sind, hier einige Bezugsquellen:
Typ AP 119/G 20polig (deshalb 2 Stück)
Firma Ratev, Lochnerstr. 1, 4030 Ratingen
oder
Typ K5 50polig (deshalb nur 1 Stück)
Best.Nr.: 58 F 800
Firma Bürklin, Kölner Str. 42, 4000 Düsseldorf

Tabelle 1. Stückliste zur Umschaltplatine

Auf zu neuen Welten

Der parallele IEEE-Bus ist eine der wichtigsten Verbindungen der »großen« Commodore-Computer zur Außenwelt. Mit unserem Selbstbau-Interface eröffnen Sie Ihrem C 64 die gesamte Palette der Commodore-Peripherie.

Ohne Zweifel, es lohnt sich! Speicherkapazitäten von mehreren MByte, beschleunigte Lade- und Speichervorgänge sowie eine rasante Dateiverarbeitung machen die »großen« Peripheriegeräte, wie das Diskettenlaufwerk CBM 4040, interessant. Das einzige Problem ist die im C 64 eingebaute Schnittstelle. Der serielle IEEE-488-Bus (auch als IEC-Bus bezeichnet) ist nicht zum Anschluß dieser Geräte geeignet. Aber der C 64 wäre nicht der C 64, der seinen weltweiten Siegeszug angetreten hat, wenn man dieses Hindernis nicht überwinden könnte. Unser Selbstbau-Interface verwandelt den Expansion-Port in eine IEEE-488-Schnittstelle, wie sie von den CBM-Computern her bekannt ist.

Dabei sind im wesentlichen zwei Hindernisse zu überwinden. Einerseits müssen die Daten des C 64 zwischengespeichert und aufbereitet werden. Andererseits muß dem C 64 mitgeteilt werden, daß er seine Daten nicht mehr wie gewohnt seriell, sondern parallel ausgibt. Die Hauptaufgabe der Bedienung des IEEE-Busses übernimmt dabei ein 6821-Schnittstellenbaustein. Mehrere Gründe sprechen für diesen Ein-/Ausgabebaustein. Da ist zum einen der ausgesprochen günstige Preis. Er ist in nahezu jedem Elektronikversand unter 10 Mark zu haben. Ein 6526, wie er im C 64 sogar gleich zweimal vorhanden ist, kostet dagegen zirka 45 Mark. Zum anderen ist der 6821, im Gegensatz zum ebenfalls möglichen 6522, ohne Probleme an den C 64 anschließbar. Der 6522 verträgt das Timing des C 64 nicht und benötigt deshalb noch eine Zusatzschaltung, die das Taktsignal des C 64 etwas verzögert. Der 6821 läßt sich übrigens durch die stromsparende CMOS-Variante 6321 problemlos ersetzen.

Die Daten werden vom C 64 allerdings nicht direkt auf den IEEE-Bus, sondern über einen Bustreiber 74LS245 geführt. Dies ist nötig, um den 6821 (6321) vor Beschädigung zu schützen und zugleich den sicheren Betrieb von mehreren Geräten am Bus zu gewährleisten. Die Steuersignale werden dabei über den Inverter-Baustein 4069 geführt.

Die Realisierung

Der verwendete PIA (Parallel Interface Adapter)-Baustein 6821 enthält zwei Ports. Jeder Port verfügt über drei Register: das Dateneingabe- und Datenausgaberegister, das Datenrichtungsregister und das Control-/Statusregister. Diese Register wurden auf einige bisher unbenutzte Register des C 64 gelegt. Im einzelnen befinden sich diese in folgenden Speicherzellen:

$DE00	Datenregister A/Datenrichtungsregister A
$DE01	Controllregister A
$DE02	Datenregister B/Datenrichtungsregister B
$DE03	Controllregister B

Die Unterscheidung zwischen dem Datenrichtungs- und Ein-/Ausgaberegister erfolgt über das Bit 2 des Controllregisters. Ist es gelöscht (0), dann ist das Datenrichtungsregister selektiert. Im Datenrichtungsregister ist jeder Ein-/Ausgabeleitung ein Bit zugeordnet. Der Zustand dieses Bits legt fest, ob die betreffende Leitung Eingang oder Ausgang ist. Ist das Bit gesetzt, so ist die zugeordnete Leitung Ausgang, ist das Bit gelöscht, so wird die Leitung als Eingang geschaltet. Beim IEEE-488-Modul sind diese Leitungen wie in Bild 1 abgebildet, geschaltet.

Natürlich kann man das IEEE-Modul auch als eine Art Userport-Erweiterung benutzen. In der Regel wird man aber Geräte wie Disket-

TRW-Stecker	Signal	Amphenol-Stecker
Pin 1	DIO 0	Pin 1
Pin 2	DIO 1	Pin 2
Pin 3	DIO 2	Pin 3
Pin 4	DIO 3	Pin 4
Pin 5	EOI	Pin 5
Pin 6	DAV	Pin 6
Pin 7	NRFD	Pin 7
Pin 8	NDAC	Pin 8
Pin 9	IFC	Pin 9
Pin 10	--	Pin 10
Pin 11	ATN	Pin 11
Pin 12	GND	Pin 12
Pin A	DIO 4	Pin 13
Pin B	DIO 5	Pin 14
Pin C	DIO 6	Pin 15
Pin D	DIO 7	Pin 16
Pin E bis N	GND	Pin 17 bis 24

Bild 2. Verdrahtungsschema für ein Peripheriekabel

PA0 bis PA7	Ein-/Ausgang D0 bis D7
CA2	steuert Datenrichtung des 74LS245
PB0	EOI (End Or Identify) Ausgang
PB1	EOI (End Or Identify) Eingang
PB2	DAV (Data Valid) Eingang
PB3	DAV (Data Valid) Ausgang
PB4	NRFD (Not Ready For Data) Eingang
PB5	NRFD (Not Ready For Data) Ausgang
PB6	NDAC (No Data Accepted) Eingang
PB7	NDAC (No Data Accepted) Ausgang
CB2	ATN (Attention) Ausgang

Bild 1. Beschaltungsplan des IEEE-488-Interfaces

Achtung! Erst mit CTRL und + einschalten

@	= Auslesen des Fehlerkanals
@$	= Directory ohne Programmverl.
@N,I,R,S,O	= New, Initialize, Rename,Scratch, Copy
/Name	= Laden von Programmen
↑Name	= Laden und starten
←Name	= Speichern von Programmen

Auch Joker und Fragezeichen im Namen sind erlaubt.

Bild 3. Auch die DOS-5.1-Befehle kennt das IEEE-Interface

tenlaufwerke (4040, 8050, 8250, 2031, SFD1001) oder aber Drucker und Meßgeräte anschließen. Dazu benötigt man noch ein spezielles Anschlußkabel, das auf der Computerseite mit einem TRW 251-12-50-170 und auf der Peripherieseite mit einem 24poligen Amphenol-Stecker verbunden ist. Die Anschlußbelegung ist im Verdrahtungsschema (Bild 2) abgebildet.

Die Software

Wie bei fast allen Erweiterungen des Betriebssystems trat auch beim IEEE-488-Interface das Problem des Speicherbereichs auf. Die Lösung sollte sich mit fertiger Software bestens vertragen und trotzdem sofort nach dem Einschalten zur Verfügung stehen. Anstelle der Kassettenroutinen wurde die Treibersoftware für das Interface einprogrammiert. Da dort aber noch Platz war, haben wir das von der Test/Demo-Diskette bekannte DOS 5.1 implementiert. Die zusätzlichen Befehle dieser DOS-Erweiterung sind in Bild 3 zusammengefaßt. Um aber Probleme mit fertiger Software auszuschließen, muß das DOS 5.1 mit den Tasten CTRL und + eingeschaltet werden.

Der wesentlichste Grund, warum sich Diskettenlaufwerke wie die sehr gute SFD 1001 bisher nicht durchsetzen konnten, ist einerseits der Preis und andererseits das vom 1541-Floppy-Laufwerk unterschiedliche Aufzeichnungsformat. Kaum ein Anbieter professioneller Software konnte sich bisher dazu entschließen, seine Programme auch in diesem Format anzubieten. Durch einen kleinen Trick ist es möglich, mit unserem Interface Programme vom 1541 auf das 8250-Format (das SFD-1001-Laufwerk ist praktisch ein halbiertes 8250-Laufwerk) zu überspielen. Die Speicherzelle 2 des C 64 wurde zum Steuerregister umfunktioniert. Im Normalfall steht dort eine 4. Das bedeutet, daß der Drucker auch mit dem IEEE-488-Interface weiterhin seriell betrieben werden kann. Und nun der Trick: Schreibt man in Speicherzelle 2 eine 8, so wird nicht das parallele Diskettenlaufwerk, sondern die 1541 angesprochen. Stellt man die Geräteadresse des parallelen Laufwerks auf 9, so kann man mit jedem Kopierprogramm für zwei Laufwerke vom Laufwerk 8 (1541) auf Laufwerk 9 (zum Beispiel SFD 1001) Programme übertragen. Die normale Geräteadresse (ohne Umstellung) des parallelen Laufwerks ist aber weiterhin die 8. Natürlich können Sie auf diese Weise auch jede andere erlaubte Geräteadresse seriell ansprechen, Sie brauchen dazu nur die Geräteadresse in Speicherzelle 2 zu schreiben (Beispiel: POKE 2,8 um das 1541-Laufwerk anzusprechen).

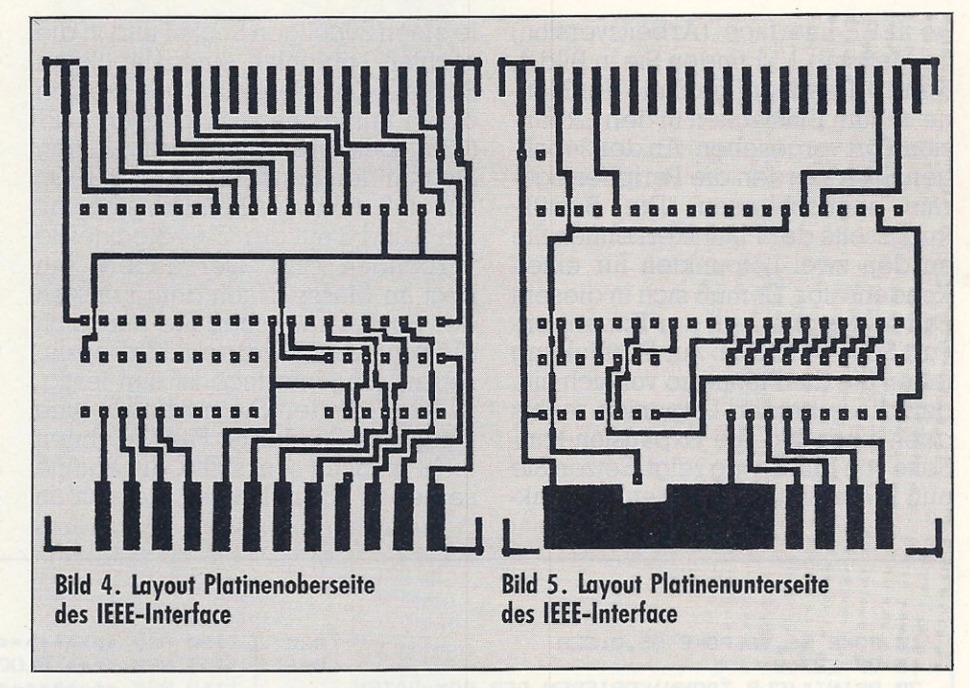
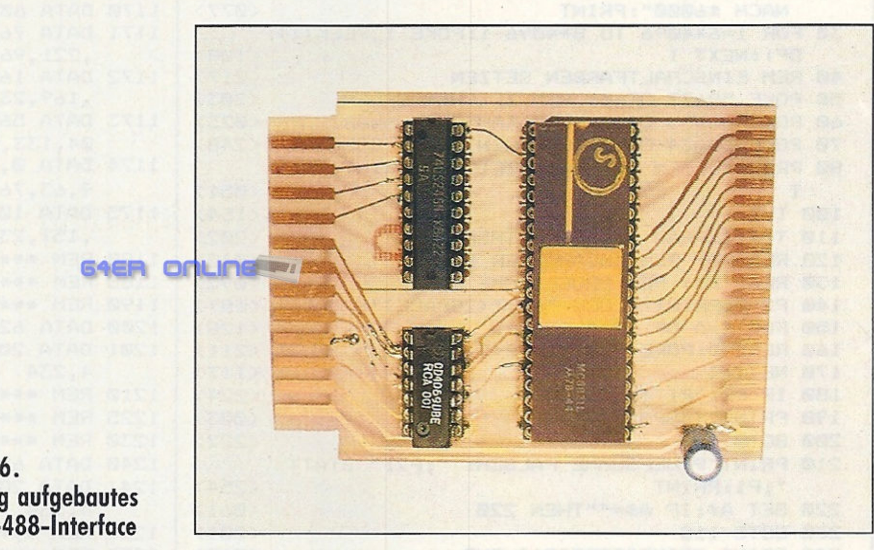

Bild 4. Layout Platinenoberseite des IEEE-Interface

Bild 5. Layout Platinenunterseite des IEEE-Interface


Bild 6. Fertig aufgebautes IEEE-488-Interface

Das Listing

Die Software (Listing 1) wird im Overlay-Verfahren auf das Original-Kernal kopiert. Dazu wird vom Programm zunächst das Betriebssystem in den Bereich ab $6000 verschoben. In einer POKE-Schleife werden die notwendigen Veränderungen vorgenommen. Nach Beendigung des Programmablaufs befindet sich im Speicherbereich des C 64 ab $6000 bis $8000 das komplett neue Kernal. Sie können es nun entweder direkt mit einem Monitor abspeichern (was zu empfehlen ist) oder aber gleich in ein 2764-EPROM brennen. Dieses EPROM wird dann mit einem Adapter anstelle des Kernal-ROMs in den U4-Steckplatz des C 64 eingebaut. Die Herstellung eines solchen Adapters haben wir in Ausgabe 4/85 ausführlich beschrieben. Es werden aber auch von verschiedenen Herstellern ähnliche Adapter angeboten.

Wie erhalte ich meine Platine?

Das Platinen-Layout und das ferti-

1	IC 4069 Inverter
1	IC 74LS245 Bustreiber
1	IC 6821 Ein-/Ausgabebaustein ersatzweise:
1	IC 6321
1	Sockel 20polig, gedreht
1	Sockel 14polig, gedreht
1	Sockel 40polig, gedreht
1	Kondensator 10 µF

Bild 7. Diese Bauteile benötigt man für das IEEE-Interface

ge IEEE-Interface (Arbeitsversion) im Maßstab 1 : 1 finden Sie in Bild 4, 5 und 6. Das 44polige Ende der Platine ist zum Einstecken in den Expansion-Port vorgesehen. An der 24poligen Seite werden die Peripheriegeräte angeschlossen. Die Bestükkungsseite der Platine erkennen Sie an den zwei Lötpunkten für einen Kondensator. Er muß sich in diesem Fall rechts oben auf der Expansion-Port-Seite befinden. Zur Bestückung legen Sie die Platine so vor sich hin, daß diese beiden Lötpunkte rechts oben sind, und die Expansion-Port-Seite von Ihnen weg zeigt. Setzen Sie nun in die linken vorderen Lötpunkte einen 20poligen Sockel und in die rechten Lötpunkte einen 14poligen Sockel. Die Sockelkerben zeigen dabei immer nach links. Den 6821 (6321) sockeln Sie analog dazu. Löten Sie nun den Kondensator (10 μ F) so ein, daß +5 V des Kondensators mit Pin 2 und 3 auf der C 64-Steckleiste verbunden sind. Der äußere Pin liegt an Masse. Nach dem Einlöten der Sockel brauchen Sie nur noch die Bausteine (Kerbe auf Kerbe) einsetzen. Das Interface ist nun fertig. In Bild 7 finden Sie nochmals eine komplette Stückliste. Falls es Ihnen nicht möglich sein sollte, die Platine selber zu ätzen, können Sie sich an die unten angegebene Adresse wenden, dort hilft man Ihnen weiter. Unter dieser Adresse ist die Leerplatine (35 Mark), der Bausatz inklusive EPROM (75 Mark) und das Fertiggerät inklusive EPROM (98 Mark) zu haben.

Noch ein paar Tips zum Abschluß: Überprüfen Sie alle Kontakte auf Durchgang. Stecken Sie das Modul nur in den ausgeschalteten Computer und vergessen Sie den Adapter für das Kernal-EPROM nicht.

(Ernst Schöberl/Arnd Wängler/ah)

Info: Frank u. Hoffmann, Wotanstr. 9, 8000 München, Tel.: 089/1782546

```
10 POKE 56,96:POKE 55,0:CLR                      <020>
15 OF=32768                                      <096>
20 PRINT"{CLR,3DOWN}KOPIEREN DER ROM-DATEN
   NACH $6000":PRINT                             <077>
30 FOR I=6*4096 TO 8*4096-1:POKE I,PEEK(I+
   OF):NEXT I                                    <108>
40 REM EINSCHALTFARBEN SETZEN                    <217>
50 POKE 58677-OF,5: REM ZEICHEN GRUEN            <203>
60 POKE 60633-OF,0: REM RAHMEN SCHWARZ           <023>
70 POKE 60634-OF,0: REM HINTERGR. SCHWARZ        <248>
80 PRINT"LESEN DER DATA-ZEILEN":PRINT:PRIN
   T                                             <051>
100 T=0                                          <154>
110 T=T+1:READ A:IF A=0 THEN 240                 <202>
120 READ B: REM ANZAHL DER BYTES                 <042>
130 READ P1: REM PRUEFSUMME                      <092>
140 P2=0:PRINT"BLOCK ";T;"{2SPACE}";             <187>
150 FOR I=A-OF TO A-OF-1+B                       <128>
160 READ D:POKE I,D:P2=P2+D                      <211>
170 NEXT I                                       <117>
180 IF P2<>P1 THEN 210                           <224>
190 PRINT"{3SPACE}OK"                            <053>
200 GOTO 110                                     <227>
210 PRINT"PRUEFSUMME FALSCH: ";P2;" STATT
    ";P1:PRINT                                   <254>
220 GET A$:IF A$=""THEN 220                      <061>
230 GOTO 110                                     <001>
240 PRINT:PRINT"FERTIG!":END                     <060>
1000 REM *****************                       <068>
1005 REM **** BLOCK 1 ****                       <107>
1010 REM *****************                       <078>
1020 DATA 57817,2,170                            <176>
1025 DATA 162,8                                  <133>
1030 REM *****************                       <099>
1035 REM **** BLOCK 2 ****                       <139>
1040 REM *****************                       <109>
1050 DATA 57895,2,170                            <213>
1055 DATA 162,8                                  <163>
1060 REM *****************                       <129>
1065 REM **** BLOCK 3 ****                       <170>
1070 REM *****************                       <139>
1080 DATA 58858,3,529                            <252>
1081 DATA 76,202,251                             <177>
1090 REM *****************                       <159>
1095 REM **** BLOCK 4 ****                       <201>
1100 REM *****************                       <169>
1110 DATA 60686,3,390                            <014>
1111 DATA 76,64,250                              <164>
1120 REM *****************                       <189>
1125 REM **** BLOCK 5 ****                       <232>
1130 REM *****************                       <199>
1140 DATA 60576,1,130                            <032>
1141 DATA 130                                    <144>
1150 REM *****************                       <219>
1155 REM **** BLOCK 6 ****                       <007>
1160 REM *****************                       <229>
1170 DATA 60859,91,11666                         <235>
1171 DATA 76,67,251,76,28,251,41,247,141,0
     ,221,96,133,149,76,39,251,120,32            <125>
1172 DATA 160,238,32,190,237,32,133,238,32
     ,169,238,48,251,88,96,36,148,48,5           <199>
1173 DATA 56,102,148,208,5,72,32,239,250,1
     04,133,149,24,96,234,76,250,250,173         <015>
1174 DATA 0,221,9,8,141,0,221,169,95,44,16
     9,63,76,6,251,32,190,237,138,162            <124>
1175 DATA 10,202,208,253,170,32,133,238,76
     ,151,238,76,81,251                          <195>
1180 REM *****************                       <249>
1185 REM **** BLOCK 7 ****                       <038>
1190 REM *****************                       <003>
1200 DATA 62637,11,1586                          <205>
1201 DATA 201,4,176,7,169,8,133,186,234,23
     4,234                                       <101>
1210 REM *****************                       <023>
1225 REM **** BLOCK 8 ****                       <079>
1230 REM *****************                       <043>
1240 DATA 62959,11,1586                          <252>
1241 DATA 201,4,176,7,169,8,133,186,234,23
     4,234                                       <141>
1250 REM *****************                       <063>
1255 REM **** BLOCK 9 ****                       <110>
1260 REM *****************                       <073>
1270 DATA 62999,3,296                            <188>
1272 DATA 32,12,252                              <056>
1280 REM *****************                       <136>
1285 REM **** BLOCK 10 ****                      <181>
1290 REM *****************                       <146>
1300 DATA 64064,192,22379                        <155>
1301 DATA 32,21,252,48,6,32,164,240,76,17,
     237,72,36,148,16,10,56,102,163,32           <031>
1302 DATA 110,250,70,148,70,163,104,133,14
     9,169,86,141,2,222,169,4,44,2,222           <030>
1303 DATA 240,251,169,60,141,3,222,169,86,
     36,163,16,2,169,87,141,2,222,173            <252>
1304 DATA 2,222,41,80,201,80,240,86,169,48
     ,141,1,222,169,255,141,0,222,169            <237>
1305 DATA 52,141,1,222,165,149,73,255,141,
     0,222,169,16,44,2,222,240,249,169           <037>
1306 DATA 8,13,2,222,141,2,222,169,255,141
     ,7,221,169,25,141,15,221,169,0,141          <068>
1307 DATA 7,221,173,7,221,240,17,44,2,222,
     80,246,169,86,141,2,222,169,255,141         <138>
1308 DATA 0,222,24,96,44,133,2,16,10,32,22
     5,255,208,209,240,3,169,128,44,169          <087>
1309 DATA 3,32,28,254,88,24,144,220,133,14
     9,32,110,250,169,52,141,3,222,96            <245>
```



Listing 1. Treibersoftware für das IEEE-488-Interface. Beachten Sie bitte die Eingabehinweise auf Seite 77.

```
1310 DATA 32,21,252,48,3,76,64,237,76,110,
     250,32,21,252,48,18,120              <056>
1320 REM *******************              <176>
1325 REM **** BLOCK 11 ****               <222>
1330 REM *******************              <186>
1340 DATA 64256,256,31864                 <198>
1341 DATA 32,142,238,76,243,237,32,21,252,
     48,9,32,17,237,76,3,238,169,95,44    <102>
1342 DATA 169,63,32,75,250,76,233,250,32,2
     1,252,48,248,173,0,221,76,193,237    <092>
1343 DATA 32,21,252,48,6,32,54,237,76,204,
     237,133,149,32,110,250,169,246,141   <128>
1344 DATA 2,222,169,52,141,3,222,24,96,32,
     21,252,48,6,32,54,237,76,190,237     <031>
1345 DATA 76,228,250,32,21,252,48,6,120,16
     9,0,76,22,238,169,48,141,1,222,169   <138>
1346 DATA 0,141,0,222,169,60,141,1,222,169
     ,214,141,2,222,169,255,141,7,221     <004>
1347 DATA 169,25,141,15,221,169,0,141,7,22
     1,173,7,221,240,50,169,4,44,2,222    <063>
1348 DATA 208,244,169,246,141,2,222,173,2,
     222,74,74,176,5,169,64,32,28,254     <050>
1349 DATA 173,0,222,73,255,72,169,118,141,
     2,222,169,4,44,2,222,240,251,169     <031>
1350 DATA 246,141,2,222,104,24,96,44,133,2
     ,16,5,32,237,246,208,176,169,2,32    <079>
1351 DATA 28,254,169,13,72,208,228,201,131
     ,208,3,76,238,229,201,130,208,6,32   <135>
1352 DATA 44,247,76,254,229,76,254,229,70,
     148,162,56,142,1,222,162,48,142,3    <108>
1353 DATA 222,169,169,141,2,222,169,255,14
     1,0,222,162,52,142,1,222,162,52,142  <170>
1354 DATA 3,222,141,0                     <229>
1360 REM *******************              <216>
1365 REM **** BLOCK 12 ****               <007>
1370 REM *******************              <226>
1380 DATA 64512,40,5057                   <123>
1381 DATA 222,169,86,141,2,222,173,0,222,7
     6,142,238,165,194,133,173,165,193    <130>
1382 DATA 133,172,96,72,165,186,41,15,197,
     2,208,5,104,44,29,252,96,104,44,28   <186>
1383 DATA 252,96                          <035>
1390 REM *******************              <246>
1395 REM **** BLOCK 13 ****               <038>
1400 REM *******************              <000>
1410 DATA 64999,9,1113                    <118>
1411 DATA 169,4,133,2,169,64,76,243,253   <178>
1420 REM *******************              <020>
1425 REM **** BLOCK 14 ****               <069>
1430 REM *******************              <030>
1440 DATA 65405,3,549                     <089>
1441 DATA 76,222,251                      <028>
1450 REM ********************************  <084>
1460 REM ********************************  <094>
1470 REM WENN OHNE DOS 5.1, DANN NUR      <112>
1480 REM NOCH ZEILE 10000 EINGEBEN!!      <109>
1490 REM ********************************  <124>
1500 REM ********************************  <134>
1520 REM *******************              <120>
1525 REM **** BLOCK 15 ****               <170>
1530 REM *******************              <130>
1540 DATA 63276,212,27895                 <145>
1541 DATA 76,105,247,248,248,248,248,248,2
     49,247,247,247,249,248,169,169,169   <151>
1542 DATA 169,169,0,207,207,207,6,243,37,4
     7,173,94,174,95,62,177,64,35,81,0    <054>
1543 DATA 13,68,79,83,32,86,53,46,49,32,65
     ,75,84,73,86,73,69,82,84,13,0,76     <027>
1544 DATA 123,247,162,2,189,102,247,149,12
     4,202,16,248,169,8,141,145,3,76,211  <131>
1545 DATA 249,133,166,134,167,186,189,1,1,
     201,230,240,4,201,140,208,23,189     <229>
1546 DATA 2,1,201,167,240,4,201,164,208,12
     ,165,166,162,8,221,69,247,240,17     <221>
1547 DATA 202,16,248,165,166,166,167,201,5
     8,176,3,76,128,0,76,138,0,134,165    <047>
1548 DATA 141,148,3,32,43,249,166,165,169,
     62,133,187,169,3,133,188,173,145     <010>
1549 DATA 3,133,186,189,47,247,72,189,58,2
     47,72,96,152,240,69,162,9,189,69     <046>
1550 DATA 247,240,8,205,62,3,240,13,232,16
     ,243,173,62,3,201,36,240,81,76,249   <070>
1551 DATA 247,198,183,169,64,133,187,169,3
     ,133,188,76,199,247,165,186,32,177   <151>
1552 DATA 255,169,111                     <192>
1560 REM *******************              <161>
1565 REM **** BLOCK 16 ****               <212>
1570 REM *******************              <171>
1580 DATA 63488,256,34255                 <186>
1581 DATA 133,185,32,147,255,160,0,185,62,
     3,32,168,255,200,196,183,144,245     <022>
1582 DATA 32,174,255,76,55,248,165,186,32,
     180,255,169,111,133,185,32,150,255   <137>
1583 DATA 32,165,255,201,13,240,6,32,22,23
     1,76,36,248,32,22,231,32,171,255     <252>
1584 DATA 76,121,0,169,96,133,185,32,213,2
     43,165,186,32,180,255,165,185,32     <034>
1585 DATA 150,255,169,0,133,144,160,3,132,
     183,32,165,255,133,195,32,165,255    <068>
1586 DATA 133,196,164,144,208,62,164,183,1
     36,208,235,166,195,165,196,32,205    <097>
1587 DATA 189,169,32,32,22,231,32,165,255,
     166,144,208,38,201,0,240,24,32,22    <058>
1588 DATA 231,32,225,255,240,26,32,228,255
     ,240,232,201,32,208,228,32,228,255   <108>
1589 DATA 240,251,208,221,169,13,32,22,231
     ,160,2,76,81,248,32,66,246,169,13    <068>
1590 DATA 32,22,231,76,121,0,166,43,164,44
     ,173,148,3,201,37,208,3,169,1,44     <008>
1591 DATA 169,0,133,185,169,0,32,213,255,1
     76,46,173,148,3,201,37,240,25,165    <077>
1592 DATA 175,133,46,165,174,133,45,32,89,
     166,32,51,165,173,148,3,201,173,240  <184>
1593 DATA 4,201,47,208,3,76,134,227,169,0,
     32,144,255,32,142,166,76,174,167     <035>
1594 DATA 76,134,227,162,2,189,171,227,149
     ,124,202,16,248,76,134               <063>
1600 REM *******************              <201>
1605 REM **** BLOCK 17 ****               <253>
1610 REM *******************              <211>
1620 DATA 63744,256,30574                 <221>
1621 DATA 227,32,89,225,76,24,248,164,183,
     185,62,3,41,15,141,145,3,136,240     <063>
1622 DATA 20,185,62,3,41,15,168,240,12,173
     ,145,3,24,105,10,136,208,251,141     <028>
1623 DATA 145,3,76,121,0,160,0,32,115,0,17
     0,208,3,76,194,249,169,96,133,124    <097>
1624 DATA 165,122,72,165,123,72,138,201,34
     ,240,30,32,115,0,208,247,104,133     <032>
1625 DATA 123,104,133,122,32,121,0,162,0,2
     01,34,240,14,162,2,228,123,208,92    <058>
1626 DATA 162,0,240,9,104,104,162,0,32,115
     ,0,240,86,201,34,240,82,201,61,240   <108>
1627 DATA 4,201,58,208,2,162,255,201,91,24
     0,10,153,62,3,141,147,3,232,200,16   <122>
1628 DATA 223,32,115,0,240,46,141,146,3,32
     ,115,0,240,38,201,93,208,34,224,16   <124>
1629 DATA 176,30,173,147,3,201,42,208,5,13
     6,202,169,63,44,169,32,224,15,176    <113>
1630 DATA 7,153,62,3,200,232,16,245,173,14
     6,3,208,195,162,76,134,124,76,8,175  <218>
1631 DATA 132,183,162,76,134,124,32,121,0,
     240,5,32,115,0,208,251,96,162,0,189  <191>
1632 DATA 81,247,240,6,32,22,231,232,208,2
     45,96,0,133,146,228,215,208,15,138   <159>
1633 DATA 208,160,165,169,48,189,201,16,14
     4,185,133,150,176,181,138,69,155     <097>
1634 DATA 133,155,165,180,240,210         <087>
10000 DATA 0                              <234>
10010 REM ****** ENDE ******              <228>
```


Listing 1. Treibersoftware (Schluß)

Neue Spieletests – so wird bewertet

Die 64'er-Redaktion hat ein neues Konzept für Spieletests entwickelt. In jeder Ausgabe werden wir etwa fünf aktuelle Spiele testen. Dabei vergeben wir in fünf Kategorien jeweils bis zu 15 Punkte. Die Punktzahlen finden Sie in Balkendiagrammen am Schluß eines Tests. Da nur über dem Durchschnitt liegende Spiele vorgestellt werden sollen, fangen unsere Skalen erst bei fünf Punkten an. Zu den Kategorien selber: Mit Spielidee bezeichnen wir die Handlung des Spiels, seinen Spielwitz, seine Hintergrundstory und seine Originalität. Die Punkte Grafik und Sound brauchen wohl nicht weiter erläutert werden. Damit Sie eine direkte Vergleichsmöglichkeit haben, wurden zwei Spiele ausgesucht, deren Grafik oder Sound wir derzeit für das Beste halten: Bei der Grafik wäre das »Impossible Mission«, beim Sound »Ghostbusters«. Beiden geben wir auf dem entsprechenden Gebiet 11 Punkte. Warum so wenig, wenn wir sie doch für die besten halten? Nun, die Noten müssen steigerungsfähig bleiben. Gerade die für die nächsten Monate angekündigten Neuerscheinungen sollen nämlich mit bisher ungeahnter Grafik und Sound auftrumpfen. Weiter geht's mit dem Punkt Schwierigkeitsgrad. Hier eine Note zu vergeben, fällt schwer, ändert sich der Schwierigkeitsgrad bei den meisten Spielen doch laufend. Wir wollen hier überschlagsmäßig angeben, für wie schwer man das Spiel halten darf. Unlösbare Spiele erhalten 15 Punkte. Auch hier vielleicht ein Beispiel: »Championship Loderunner« würde 14 Punkte erhalten. Als letztes noch die Motivation. Wie lange wird man wohl Spaß an dem Spiel haben. Einen Nachmittag oder mehrere Monate? Motivation bedeutet aber auch, wie schwer es einem fällt, sich nachts um 2 Uhr endlich vom Spiel loszureißen und ins Bett zu gehen.

Damit die Bewertungen einerseits möglichst objektiv, andererseits von Ausgabe zu Ausgabe immer vergleichbar sind, werden sie jeden Monat von einem festen Team, bestehend aus Redaktionsmitgliedern und freien Mitarbeitern, ausdiskutiert. Wir halten uns dabei nicht für unfehlbar, glauben aber, faire Entscheidungen treffen zu können.

(B. Schneider/M. Kohlen/rg)

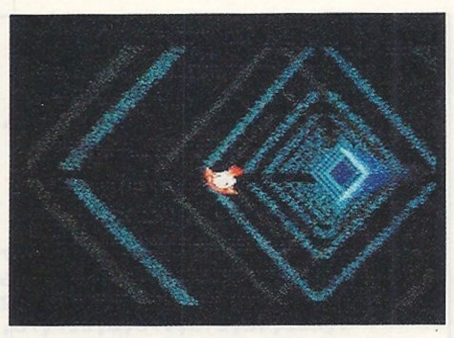
Trends und Flops

Auch diese Spalte wird vorläufig zu einer festen Einrichtung werden. Hier werden wir über Trends auf dem Spielesektor, aber auch über Flops berichten.

Nun gleich hinein ins Getümmel: Lange erwartet und zum Redaktionsschluß immer noch nicht lieferbar waren die US-April-Neuerscheinungen der Firmen Electronic Arts, Broderbund und Epyx. Schon seit Wochen kursieren Gerüchte um Programme wie »Summergames II«, »Karateka«, »Rescue on Fractalus«, »Skyfox« und einige andere mehr. Teilweise handelt es sich hier um Umsetzungen von Apple- oder Atari-Programmen, von deren hoher Qualität man sich schon überzeugen konnte. Auch von den anderen Neuerscheinungen dieser Firmen hört man bis jetzt nur Gutes. Wir bleiben am Ball und versuchen, so schnell wie möglich über diese Spiele zu berichten. »Mail Order Monsters« und »Racing Destruction Set« erreichten uns erst nach Redaktionsschluß und werden in der 64'er-Ausgabe 8 besprochen.

Um beim Thema Trends zu bleiben: Daß man heutzutage mit primitiven Schießspielen keinen Hund mehr hinterm Ofen hervorlocken kann, sollten die Softwarehersteller inzwischen begriffen haben. Und auch ein einfaches Jump-and-Run-Spiel, das keinerlei neue Ideen mit sich bringt, wird wohl nur wenige Käufer finden. Trotzdem tauchen da immer noch Spiele wie »Super Zaxxon« auf. »Super Zaxxon« ist grafisch schlechter als das echte »Zaxxon«, hat aber dafür mehr unterschiedliche Stufen. Langeweile ist allerdings vorprogrammiert, denn am Spielablauf (Ballern, Ballern, Ballern ...) hat sich überhaupt nichts geändert. Weiteres Beispiel ist »Conan«. Wer nun einen Adrenalinspiegelheber erwartet, der wird von einem einfallslosen Jump-and-Run-Spiel mit einfacher, ruckhafter Grafik enttäuscht. Was hätte sich nicht alles aus einem solch gewaltigen Stoff machen lassen. Wohlgemerkt: Diese beiden Spiele sind leider keine Einzelfälle. Besonders tragisch ist nur, daß gerade zwei relativ bekannte Firmen, Hesware und Datamost, für diese Flops verantwortlich sind. Eines sei also jedem gesagt, der sich ein Spiel kaufen möchte: Immer vorher Probespielen oder Testberichte lesen, am besten beides, denn keiner gibt gern 100 Mark für nichts und wieder nichts aus.

(B. Schneider/M. Kohlen/rg)



Master Of The Lamps

In diesem Spiel von Activision geht es zu wie in einer Geschichte aus Tausendundeiner Nacht: Sie müssen als Prinz einen bösen Lampengeist in die Lampe zurückbefördern, damit Sie die Erbnachfolge antreten können. Doch leider ist die Lampe kaputt, und ihre Einzelteile sind über knapp 20 Screens verstreut. Sie müssen von Ihnen zusammengesucht werden. Von Screen zu Screen fliegt man durch einen »Zeittunnel« in schneller und gut gemachter 3D-Grafik. Am Ende des Fluges erwartet den Spieler dann der Lampengeist. Dieser wirft mit farbigen Noten, die man nur abwehren kann, wenn man schnell genug einen gleichfarbigen Gong anschlägt. Hat man diesen Angriff überstanden, gehört einem auch schon eines der Lampenteile, und weiter geht's durch den Zeittunnel. Wie schon gesagt, ist die Grafik recht schön. Besonders gut ist aber die Musik geraten: In Ghostbusters-Qualität dröhnt einem da pro Screen eine andere Melodie entgegen, die stilistisch zwar nicht zum Stil paßt (Boogie Woogie im Orient?), dafür aber der reinste Ohrenschmaus ist. Da vergißt man sogar das Spielen. Damit ist es allerdings auch schon getan. Abwechslungsreich ist das Ganze nämlich wirklich nicht. Die zwei geschilderten Situationen wiederholen sich nur mit steigendem Schwierigkeitsgrad. Schade, denn sonst wäre dies derzeit eines der besten Spiele. (B. Schneider/rg)

Titel	Master of the Lamps (5 7 9 11 13 15)
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	wenig abwechslungsreich
Hersteller	Activision
Preis	D 69 Mark, K 49 Mark
Bezugsquelle	Ariolasoft Postfach 1350 4830 Gütersloh 1

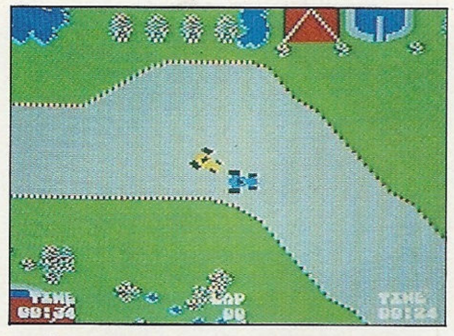


World Series Baseball

Sportspiele scheinen sich zu den Rennern des Sommergeschäfts 1985 zu entwickeln. Paradoxerweise kommt eine der besten Simulationen einer typisch amerikanischen Sportart aus England: Baseball. Wer das Spielprinzip absolut nicht kennt, der wird seine Schwierigkeiten haben. Die Anleitung geht davon aus, daß man die Spielregeln kennt. Wer aber ungefähr weiß, wie es geht, dem stehen spannende Stunden am Computer bevor, ob allein gegen den sehr spielstarken Computer oder zu zweit. Der Joystick erfüllt dabei, je nach Spielsituation, verschiedene Funktionen, sei es nun die Schlägerhaltung oder die Spielerauswahl. Neben der Aufsicht auf das gesamte Spielfeld werden auf einer Anzeigetafel die Spieler in Großaufnahme gezeigt. Die Spieler laufen trickfilmähnlich ein und übers Feld, und für die Spielpausen gibt's sogar eine Tanzgruppe. Weniger gelungen ist dagegen der Sound, der abgesehen von der amerikanischen Nationalhymne und der Begleitmusik der Tänzerinnen, auf einfache Ball- und Klatschgeräusche beschränkt ist. Wie schon erwähnt, ist der Computer recht spielstark, eher schon zu stark. Aber am meisten Spaß machen solche Spiele ja sowieso zu zweit. Für Anfänger kommt die komplizierte Bedienung erschwerend hinzu. Wer sich aber trotzdem nicht abschrecken läßt, der erhält eine der besten Mannschaftssport-Simulationen, die es derzeit gibt. (B. Schneider/rg)



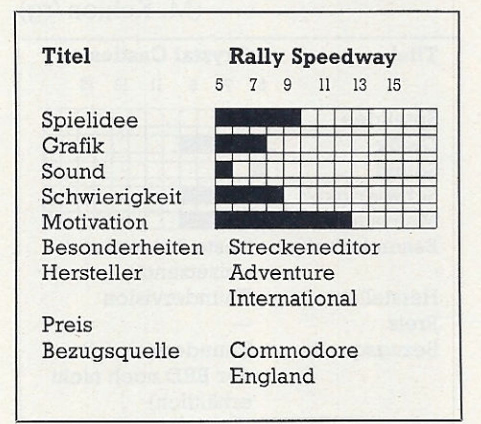
Titel	Baseball
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Computergegner oder 2 Spieler
Hersteller	Imagine
Preis	zirka 29 Mark
Bezugsquelle	Rushware An der Gümpgebrücke 24 4044 Kaarst 2



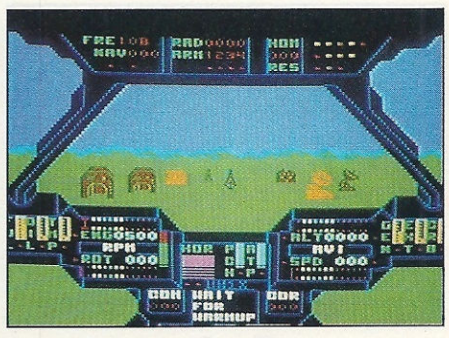
Rally Speedway

Autorenn-Simulationen gibt es inzwischen viele. Überzeugt hat uns aber ein Programm von Adventure Int. und Commodore (!). »Rally Speedway« hat keine aufwendige Grafik und keinen besonderen Sound. Die Idee eines Autorennens ist relativ alt und die Ausführung nicht überragend. Wenn zwei Spieler über die Strecken jagen, dann ist ordentlich was los. Da sich alles auf einem Screen abspielt, und nicht auf zweien wie bei Pitstop II, mußte eine Sonderregelung eingeführt werden: Wer zu weit zurückblickt, erhält fünf Strafsekunden, dann werden beide Wagen wieder Kopf an Kopf gestartet. Verursacht man eine Kollision mit einem der zahlreichen Hindernisse, gibt's gleich zehn Strafsekunden. Natürlich kann man sich gegenseitig rammen oder von der Fahrbahn drängen.

In einem Hauptmenü können sehr viele die Fahreigenschaften der Autos betreffende Parameter eingestellt werden. Dazu gehören, neben den Witterungsbedingungen, auch die Höchstgeschwindigkeit und das Beschleunigungs- und Bremsvermögen. Wem die sechs mitgelieferten Rennstrecken nicht ausreichen, der kann sich mit einem Screen-Editor ganz einfach selber welche basteln. Rally Speedway ist neben Pitstop II das interessanteste derzeit erhältliche Autorennen, vorausgesetzt, man spielt zu zweit. Der Computer hält sich selbst nämlich wohlweislich vom Lenkrad fern. (B. Schneider/rg)



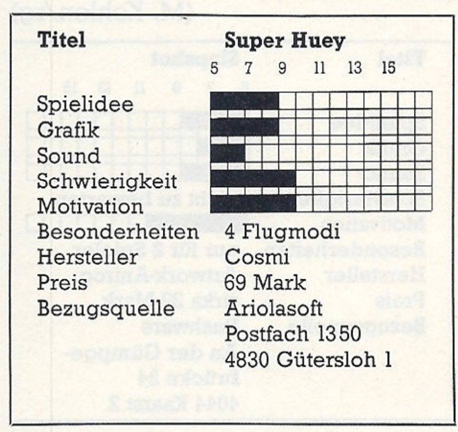
Titel	Rally Speedway
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Streckeneditor
Hersteller	Adventure International
Preis	
Bezugsquelle	Commodore England



Super Huey

Vom Programmierer dieses Spiels war man bisher nur Action gewöhnt: Paul Norman ist der Autor der bekannten Spiele »Forbidden Forest«, »Aztec Challenge« und »Caverns of Khafka«. Mit »Super Huey« wagte er sich nun in das Genre der Simulatoren. Und das mit recht gutem Erfolg. Simuliert wird das Fliegen eines »UH-1X«-Helikopters, eines der neuesten aus der Realität gegriffenen Modelle. Hat man sich durch das Anleitungsmaterial durchgekämpft, weiß man nicht nur über die Bedienung des Simulators Bescheid, sondern hat gleich ein paar Lektionen über echte Hubschrauber im klugen Köpfchen. Der reale UH-1X hat übrigens ein besonderes Stabilisationssystem, das das Fliegen wesentlich vereinfacht. Natürlich wurde das in der Simulation berücksichtigt.

Hat man den Huey erst einmal in die Lüfte erhoben, bietet sich ein eindrucksvoller 3D-Effekt. Das majestätische Schwebegefühl kann man mit vier verschiedenen Modi erleben: Flight Instruction, Exploration, Rescue und Combat. Flight Instruction soll den Spieler mit bestimmten Flugtechniken und der Bedienung vertraut machen, Exploration ist die Simulation eines Erkundungsfluges. Bei Rescue müssen Soldaten gerettet und an sichere Orte gebracht werden. Und Combat schließlich ist der aggressive Teil. Hier heißt es: Kampf den Panzern, Flaks und Feindhubschraubern. (M. Kohlen/rg)



Titel	Super Huey
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	4 Flugmodi
Hersteller	Cosmi
Preis	69 Mark
Bezugsquelle	Ariolasoft Postfach 1350 4830 Gütersloh 1



Slapshot

Sportspiel-Enthusiasten dürfen sich freuen: »Slapshot« bietet zwar nicht alles, was ein Real-Life-Eishockeyspiel an Spaß bringt, ist aber doch eines der schönsten Sportspiele, die es gibt.

Gegen den Computer kann man leider nicht spielen, aber zu zweit macht es sowieso mehr Spaß.

Slapshot erinnert sehr stark an Soccer, spielt sich aber etwas anders. In Slapshot ist Faulen erlaubt: Ein kurzer Druck auf den Feuerknopf, und der Gegner stolpert über den gemein zwischen die Beine gelegten Hockeyschläger. Und während er da verwirrt auf dem Boden sitzt, nimmt man ihm kurzerhand den Puck ab und marschiert in Richtung Tor.

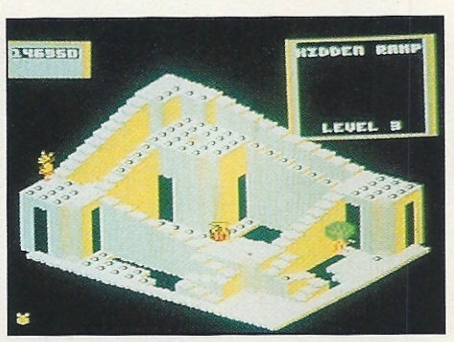
Fault man zu oft, schreit der Schiedsrichter in gut verständlichem Englisch »Penalty!« und verordnet ein Bulli. Bei einem Schuß ins Tor hört man ein begeistertes »He scored!« und wird von rauschendem Applaus (vermischt mit Pfiffen) gefeiert. Die Sprachausgabe wurde von Electronic Speech Systems produziert, die auch schon den Spielen »Impossible Mission« und »Ghostbusters« zum Erfolg verhalfen.

Wenn Slapshot gegen den richtigen Gegner gespielt wird, macht es riesig Spaß. Es kann zwar nicht ganz an das gewisse Etwas von Soccer anknüpfen, ist aber ein durchaus empfehlenswertes Spiel. Noch ein Hinweis zum Schluß. Für Joysticks mit automatischen Dauerfeuer ist dieses Spiel nicht geeignet.

(M. Kohlen/rg)



Titel	Slapshot
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	nicht zu bewerten
Motivation	
Besonderheiten	nur für 2 Spieler
Hersteller	Artwork-Anirog
Preis	zirka 29 Mark
Bezugsquelle	Rushware An der Gümpgebrücke 24 4044 Kaarst 2



Crystal Castles

»Crystal Castles« ist den Spielhöllenfreaks ein Begriff. Nachdem man die Atarisoft-Versionen für Telespiel, Atari- und Apple-Computer nicht gerade als gut bezeichnen kann, hat sich die englische Firma Thundervision des Problems angenommen und eine C 64-Version produziert. Das Ergebnis ist beeindruckend: Die Grafik ist identisch zu der des Automaten, nur die Auflösung der Sprites kommt nicht ganz an die der Spielhallenversion heran.

Bei Crystal Castles handelt es sich um ein Pac-Man-ähnliches Spiel, das auf dreidimensionalen Ebenen spielt. Held des Spiels ist ein kleiner Bär, der herumläuft und Punkte aufsammeln muß. Dabei wird er gestört durch laufende Bäume, fressende Monster und die Hexe Berthilda. Zur Stärkung gibt's dann einen Topf voll Honig und einen magischen Hut.

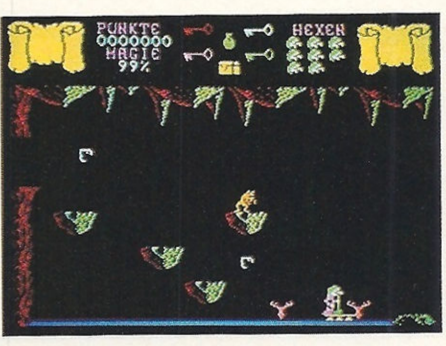
Die Güteklasse der Spielhallen- zu Computer-Adaption ist erstaunlich, denn nicht nur die Grafik, sondern auch die spielerischen Elemente gleichen sich. Die vielen kleinen Tricks, die sich auf dem Automaten anwenden lassen, sind auch an der C 64-Version benutzbar. Wer die Nummer 1 am Automaten werden will, kann also vorher auf dem Computer üben.

Crystal Castles zeichnet sich trotz der schwierigen Steuerung durch Spielwitz und gelungene Grafik aus. Der Sound ist einfach, aber durchaus passend und nicht störend. Dieses Spiel ist eine gelungene Anpassung an den Computer.

(M. Kohlen/rg)



Titel	Crystal Castles
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	beste Automatenumsetzung
Hersteller	Thundervision
Preis	—
Bezugsquelle	Thundervision (in der BRD noch nicht erhältlich)



Hexenküche

Palace Software ist das Tochterunternehmen einer Filmfirma. Dies merkt man sofort an der hervorragenden Grafik dieses Spieles.

»Hexenküche« ist in mehreren Punkten etwas Besonderes. Als erstes ist zu erwähnen, daß dieses Spiel gleichzeitig in England und Deutschland erscheint. Ein gleichzeitiger Erscheinungstermin war bisher unüblich. Als zweites ist für Deutschland und England je ein gesonderter Wettbewerb ausgeschrieben. Zu gewinnen ist ein goldener Hexenbesen (näheres siehe Happy-Computer 7/85). Als dritter Punkt ist der Text anzuführen. Für den deutschen Markt ist die englische Version übersetzt worden.

Ziel von Hexenküche ist es, eine Hexe zu ihrem goldenen Besen zu führen. Zuerst muß die Hexe durch eine Landschaft fliegen und einen Schlüssel suchen, mit dem sie in den »Untergrund« dieser Landschaft gelangen kann. Hier ist dann der goldene Besen versteckt. Doch dies hört sich einfacher an, als es ist. Auf der Suche nach dem Schlüssel wird die Hexe von Fledermäusen, Geistern, Haien und Vögeln behindert, die ihr die Kraft rauben. Der Teil, der im Untergrund spielt, ist ein Hüpf- und Springspiel, das nicht einfach zu bewältigen ist.

Alles in allem ist Hexenküche ein Spiel, das Spaß macht und nicht so schnell langweilig wird. Auch bei diesem Spiel ist der neue Trend zu niedrigen Software-Preisen zu erkennen.

(rg)

Titel	Hexenküche
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Wettbewerb
Hersteller	Palace-Software
Preis	zirka 29 Mark
Bezugsquelle	Rushware An der Gümpgebrücke 24 4044 Kaarst 2

HYPRA-ASS — Ein Assembler der Spitzenklasse

Das war noch nie da. Ein Super-Assembler zum Abtippen. Mit Leistungsmerkmalen, die Sie sich schon immer gewünscht haben, übertrifft er jeden 100-Mark-Assembler.

```
READY.
/E
50    -.MA LDAYIM (WERT)
60    -             LDA #<(WERT)
70    -             LDY #>(WERT)
80    -.RT
90    -.MA INCW (ADRESSE)
100   -             INC ADRESSE
110   -             BNE END
120   -             INC ADRESSE+1
130   -END          .RT
140   -.MA DECW (ADRESSE)
150   -             PHA
160   -             LDA ADRESSE
170   -             BNE ENDE
180   -             DEC ADRESSE+1
190   -ENDE         DEC ADRESSE
200   -             PLA
210   -.RT
220   -.MA TAB (GROESSE)
230   -LBL          .BA LBL+GROESSE
240   -.RT

READY.
```

Bild 1. Formatiertes Listing immer wieder benötigter Makros

HYPRA-ASS ist ein rein in Maschinensprache geschriebener 3-Pass-Makroassembler mit integriertem Editor für den C 64 mit Diskettenlaufwerk 1541. Er belegt etwa 6 KByte RAM-Speicher von $0801 bis $1FD7. HYPRA-ASS gehört, was Leistung und Schnelligkeit betrifft, zu einem der besten Assembler. Seine hervorstechendste Eigenschaft ist die Fähigkeit, um Befehlsfolgen, die in einem Programm immer wieder auftreten, einen Block beziehungsweise ein Makro zu legen. So definierte Befehlsfolgen können durch einen einfachen Befehl an jeder beliebigen Stelle des Programms aufgerufen werden. Diese Eigenschaft haben viele Assembler, aber nur wenige können lokale Labels beziehungsweise Sprungziele innerhalb eines Makros verarbeiten. Innerhalb eines Makros sind, wenn nicht besonders gekennzeichnet, alle Labels und Variablen lokal. Neben vielen Pseudo-Opcodes, mit denen sich zum Beispiel der Objektcode auf Diskette assemblieren läßt oder Quelltexte verketten lassen, kann auch bedingt assembliert werden. Formatierte Listings können an beliebige Geräte gesendet werden. Symboltabellen lassen sich sortiert und unsortiert ausgeben.

Quelltexte werden bei HYPRA-ASS wie normale Basic-Programme erstellt. Nach der Eingabe von RETURN erscheint die soeben eingegebene Zeile formatiert auf dem Bildschirm. Die dabei benutzten Tabulatoren können durch einfache Befehle geändert werden. Um das Editieren von Quelltexten möglichst bequem zu machen, ist in HYPRA-ASS ein spezieller Editor integriert, der den normalen Basic-Editor um viele Funktionen erweitert. Neben der automatischen Zeilennumerierung können Zeilen und Zeilenbereiche formatiert gelistet werden. Auch das oft vermißte Suchen und Ersetzen, Find und Renumber sind eine Selbstverständlichkeit für diesen Editor.

(Gert Möllmann/ah)

Label:	Sprungmarke
Bedingte Assemblierung:	Abhängig von einer Variablen können zum Beispiel Quelltextzeilen beim Assemblieren übersprungen werden. Bei unterschiedlichen Variablen erhält man folglich unterschiedlichen Objektcode.
Makro:	Befehlsfolgen, die im Quelltext häufiger benutzt werden. Das Makro wird in einer Definitionszeile mit einem Namen versehen, unter dem es im Quelltext beliebig oft aufgerufen werden kann. Trifft der Assembler auf einen so definierten Namen, so wird an diese Stelle die vorher definierte Folge von Befehlen assembliert.
n-Paß:	Für n steht eine Zahl, die die Anzahl der Assemblerläufe anzeigt. Bei Assemblern, die Labels verarbeiten können, handelt es sich in der Regel um 2-Pass-Assembler. Im ersten Pass wird eine Syntaxprüfung durchgeführt und den Labels und Variablen ein numerischer Wert (Adresse) zugewiesen. (Symboltabelle wird angelegt). Im zweiten Pass wird dann der Objektcode erzeugt.
Objektcode:	Das vom Assembler erzeugte, lauffähige Programm
Quelltext:	Das mit dem Assembler erstellte Basic- oder Text-File
Symboltabelle:	Beim Assemblieren wird jedem Label und jeder Variable ein numerischer Wert zugewiesen. Die Symboltabelle enthält eine Liste aller im Programm vorkommenden Label und Variablen mit deren Wertzuweisungen.

Bild 2. Definition der im Text auftretenden Fachbegriffe

Lebenslauf:

Ich bin 28 Jahre alt. Nach dem Abitur im Jahre 1975 begann ich ein Mathematikstudium an der Ruhr Universität Bochum. Nun, nachdem ich den Grundwehrdienst abgeleistet habe, beabsichtige ich, in die Datenverarbeitung einzusteigen und eventuell ein Informatikstudium aufzunehmen. Weitere Pläne den C 64 betreffend liegen schon vor. So ist eine Textverarbeitung schon in Angriff genommen.

Am Anfang der Entwicklung von HYPRA-ASS stand ein Editor/Assemblerpaar, das von mir Ende 1984 in Basic geschrieben wurde, um dem mangelhaften Basic des C 64 auf die Sprünge zu helfen. Doch schon bald erwies sich die Langsamkeit dieser Programme als so nervenzermürbend, daß für mich nur die Wahl bestand, entweder einen in Maschinensprache geschriebenen Assembler zu kaufen oder selbst einen zu schreiben.

Da solche Assembler recht teuer sind, und weil es eine interessante Aufgabe ist, einen Assembler zu schreiben, habe ich mich für die zweite Alternative entschieden. So entstand mit Hilfe des Basic-Assemblers eine erste Version von HYPRA-ASS, die wiederum eine zweite Version assemblierte und so weiter.

Die Entwicklung von HYPRA-ASS verlief also in einem »bootstrapping«-Verfahren, was außerdem noch den Vorteil hatte, daß eine Prüfung der Funktionstüchtigkeit des Programms schon zum großen Teil im Verfahren selbst stattfand.

(Gerd Möllmann)

Damit Sie wissen, was Sie ausgeben

Egal, ob Sie viel oder wenig Geld verdienen, auf jeden Fall hat es Vorteile, einen Überblick über seine Ausgaben zu haben. Und warum soll dazu nicht der C 64 eingesetzt werden? Mit diesem elektronischen Haushaltsbuch können Sie jederzeit Bilanz ziehen.

```
      BUCHUNGEN AUSGEBEN

GK                     1985

HACKEN                  -73.00  04. 1.85
FRISOER                 -22.00  09. 1.85
FALSCHGELD             4000.00  13. 1.85
SPEEDDOS               -250.00  14. 1.85
TURBOACCESS            -255.00  14. 1.85
HACKEN                 -124.00  15. 1.85
PKW                     200.00  15. 1.85
PC 128                 -998.00  17. 1.85
PKW                      12.23  18. 1.85
SKIANZUG                350.00  24. 1.85
MIETE                  -645.00  27. 1.85
ALIMENTE               -400.00  28. 1.85
GEHALT                 2000.00  28. 1.85
LEBENSMITTEL           -177.35  29. 1.85
                      ---------
          SUMME        3617.88

WEITER MIT TASTE
```

Bild 1. Ein Menüpunkt aus dem elektronischen Haushaltsbuch

Seit Anfang 1984 der C 64 ins Haus kam, ist die Arbeit daran zu einer Leidenschaft geworden. (Wie bei vielen anderen auch!) Soweit der Beruf es ermöglicht, verbringe ich meine Freizeit (das sind hauptsächlich die Wochenenden) mit der Entwicklung von Anwendungen am Computer. Ausgangspunkt war die Begeisterung für technische »Spielereien«. Doch bald reizte die Eigenentwicklung von Programmen. Es gibt eine tiefe Befriedigung, wenn nach stunden- oder tagelanger Arbeit ein Programmteil das erste Mal gestartet wird und nach vielleicht nur einer Korrektur zufriedenstellend läuft. Ein andermal wird stundenlang über die Lösung eines Problems getüftelt. »Gelernt« habe ich jedenfalls dadurch einiges über die Basic-Programmierung des C 64. Wenn auch häufig »Probieren über Studieren« ging.

Als Familienvater finde ich volles Verständnis bei Frau und Tochter. (Das muß mal gesagt werden!)

In der Familie liegt der Ausgangspunkt für das Programm. Die Aufzeichnung der monatlichen Einnahmen und Ausgaben erfolgte manuell. Das ließ natürlich die »Ehre« des EDV-Anwenders nicht zu. Ein entsprechendes Programm mußte her. Es sollten nicht nur die Beträge als Salden auf Konten geführt werden, sondern es sollte auch möglich sein, einzelne Buchungen wieder abzurufen. Wiederkehrende Posten sollten über die einzelnen Monate hinweg beobachtbar sein.

So entstand das »Haushaltsbuch« als eine Kombination aus sequentieller und relativer Datei: die relative zur Kontenführung (Salden); die sequentielle zur Speicherung der Kostenart, des Betrages und des Buchungsdatums als Einzelpositionen.

Von der Floppy lassen sich mit »INPUT« bekanntlich nur 88 Zeichen einlesen. Der relative Datensatz hat aber eine Länge von 141 Zeichen. Die Alternative »GET« in einer FOR-NEXT-Schleife ist sehr langsam. Dies fällt besonders bei der Bildung des Jahressaldos auf.

Es blieb nur, den INPUT-Befehl zu modifizieren. Die entsprechenden DATAs für das kleine Maschinenprogramm sind im »Haushaltsbuch« enthalten und werden in den Kassettenpuffer eingelesen.

Als Tauschroutine fand gleich die im 64'er, Ausgabe 1/85 beschriebene SWAP-Simulation Verwendung. Durch diese Funktion konnte die Produktion von »String-Müll« deutlich reduziert werden.

Ich habe mich bemüht, das Programm »wasserdicht« zu machen. Floppy und Drucker werden auf Bereitschaft geprüft. Unter normalen Bedingungen dürfte es keinen Absturz geben. Aber eine Eingabe von falschen Beträgen muß der Anwender verantworten.

(Martin Schumann/gk)

Martin Schumann

Lebenslauf

Dem Jahrgang 1949 zugehörig, und damit den »alten« Heimcomputeranwendern zuzurechnen, begegnete mir während des Studiums Anfang der siebziger Jahre die Informatik zum ersten Mal. Nur die wenigsten schenkten dem damals Beachtung. Und ich schließe mich da nicht aus.

Aber die »programmierbaren«, mathematischen Taschenrechner faszinierten mich trotzdem. 1975 konnte ich dann einen HP-25 mein Eigen nennen. Die nächsten Jahre hatte ich beruflich viel mit Zahlen und Statistiken zu tun. Dabei leistete mir der »Kleine« wertvolle Dienste.

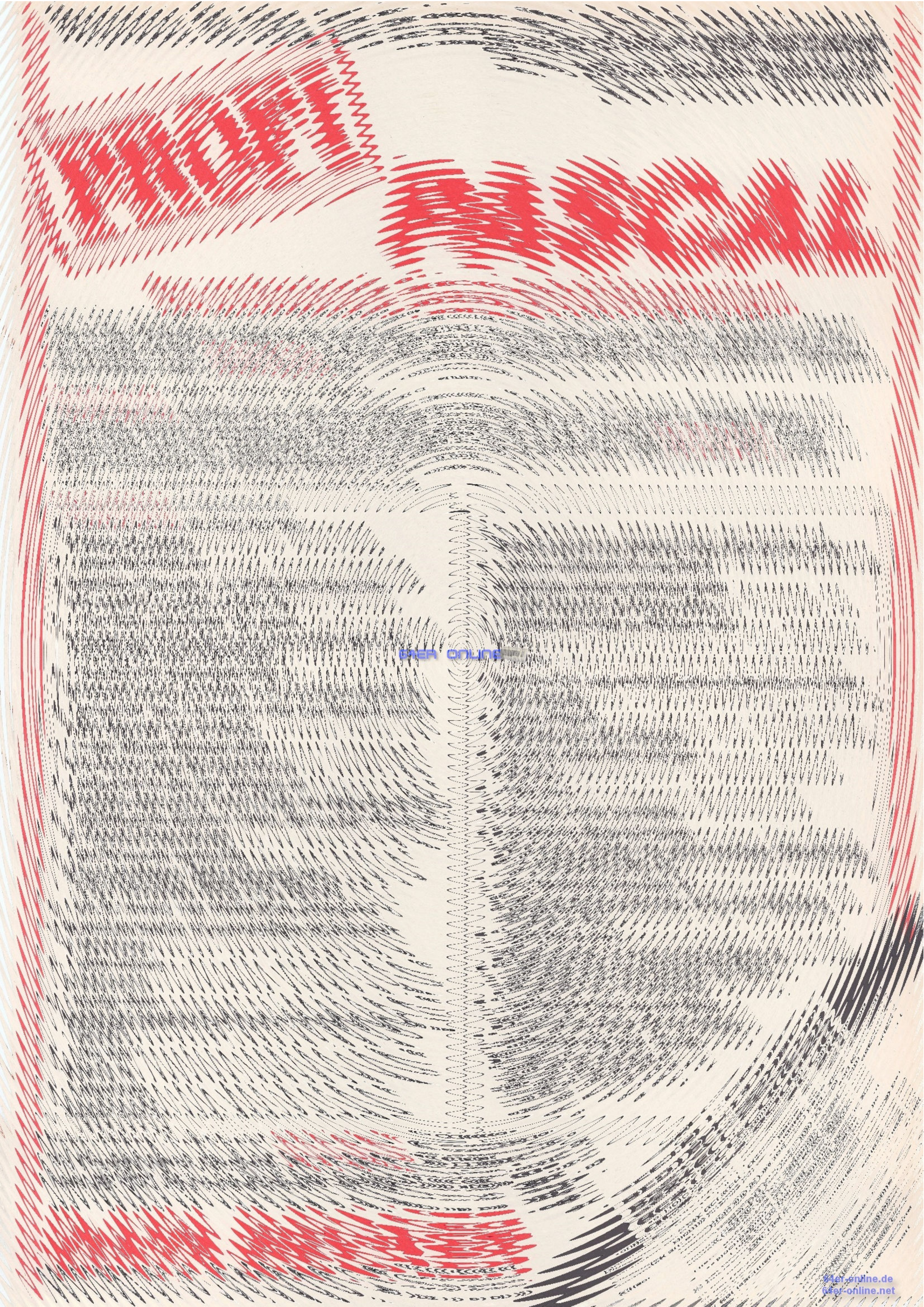
1981 änderte sich die Situation. Inzwischen beruflich mit Informationssystemen beschäftigt, wollte ich als Nutzer der EDV mehr über die eigentliche Programmierung wissen.

Lange beschäftigte mich auch die Frage nach dem »richtigen« Computer. Aufgrund der Verbreitung und der gebotenen Leistung fiel die Entscheidung auf den C 64, der Anfang 1984 schließlich gekauft wurde.

Anhand der in Zeitschriften abgedruckten Programme und diverser Bücher arbeitete ich mich in die Basic-Programmierung ein. Try and Error hieß die Devise.

Nach mehr als einem Jahr habe ich noch nicht den Spaß an der »Computer-Knobelei« und der programmtechnischen Lösung von Problemen verloren.

(Martin Schumann)



Damit Sie wissen, was Sie ausgeben

Das Programm »Haushaltsbuch« dient der Aufzeichnung der Ein- und Ausgaben im privaten Bereich. Neben den monatlichen Bestandskonten (relative Dateien) werden zur Dokumentation der Buchungsposten die Einzelpositionen in sequentiellen Buchungsdateien gespeichert.

Die in den relativen Dateien geführten Konten ermöglichen innerhalb eines Jahres den Abruf des aktuellen Standes in der »Kasse«. Aus den sequentiellen Dateien lassen sich alle Buchungen mit Angabe der Kostenart, des Betrages und des Buchungsdatums auflisten oder einzelne selektieren; zum Beispiel bestimmte Kostenarten oder Buchungsmonate.

Geschrieben wurde das »Haushaltsbuch« für einen C 64. Das für die Modifikation des INPUT-Befehls notwendige kleine Maschinenprogramm wird in dem Kassettenpuffer abgelegt. Die DATAs für die Simulation der SWAP-Tauschroutine werden in der Zero-Page in den Freiraum ab Speicherstelle 715 eingelesen.

Die sinnvolle Nutzung des Programms setzt ein Diskettenlaufwerk voraus. Ein Drucker wäre recht nützlich, ist aber keine Notwendigkeit. Sollten trotz nicht vorhandenem Drucker die betreffenden Druckroutinen aufgerufen werden, führt dies zu keinem Programmabsturz.

Ausgelegt ist das Druckunterprogramm für Epson-kompatible Drucker. Eine Anpassung an Commodore-Drucker sollte — falls es notwendig ist — leicht zu bewerkstelligen sein.

Neben der Bereitschaft des Druckers wird selbstverständlich die des Diskettenlaufwerks geprüft.

Funktionsbeschreibung

Wie aus den einleitenden Sätzen ersichtlich, gliedert sich das Programm in zwei übergreifende Programmteile: die Kontenführung und die Bearbeitung der Buchungspositionen.

Nach dem Start des Programms geben Sie bitte das zu bearbeitende Jahr (»aktuelles Jahr«) ein. Der Zeitraum wurde auf 1984 bis 2000 beschränkt. Dies kann selbstverständlich dem Bedarf des Lesers entsprechend angepaßt werden. Nur sollte dabei die Diskettenkapazität beachtet werden!

Nach Eingabe der Jahreszahl meldet sich »Haushaltsbuch« mit dem Hauptmenü. Es stehen folgende Funktionen zur Verfügung:

— 1 Konten anlegen
— 2 Buchen
— 3 Kontenübersicht
  (Kontennamen ausgeben)
— 5 Monatsübersicht
— 6 Jahresübersicht
— 7 Buchungsposten abfragen
— 0 Programm beenden

Die Funktion »Kontennamen ausgeben« wird automatisch von den übrigen Funktionen — soweit notwendig — aufgerufen. Denn warum sollte man sich die Nummern der einzelnen Konten merken, dies kann der C 64 viel besser.

Wenn Sie also »4« eingeben, erfolgt eine entsprechende Meldung. Versuchen Sie es!

Konten anlegen

Beachten Sie bei dieser Funktion, daß eine existierende Datei des gewählten Jahres gelöscht wird. Es wird zur Vorsicht abgefragt, ob Sie dies auch wirklich beabsichtigen. Beantworten Sie die Frage mit »JA«, so werden die bisherigen Konten gelöscht und Sie aufgefordert, die Zahl der neu anzulegenden Konten einzugeben. Anschließend können Sie die Konten benennen. Drücken Sie bei einer Kontonummer RETURN, so wird das Anlegen der Konten abgebrochen. Es werden nur die bis zu diesem Moment eingegebenen Konten gespeichert. Die Kontenzahl (KZ) wird entsprechend korrigiert. Geben Sie bei der Frage nach der Zahl der Konten »0« ein, so wird die Programmfunktion verlassen. Dadurch können Sie also nicht mehr benötigte Jahre löschen. (Klar, daß das auch außerhalb des Programms mit »SCRATCH« erfolgen kann.)

Gleichzeitig mit dem Anlegen der relativen (Salden-)Konten werden auch die sequentiellen Buchungsdateien eingerichtet.

Die Zahl der Konten wurde auf 20 beschränkt, um noch mit einer Bildschirmseite auszukommen. Außerdem sollten 20 Konten für die private »Buchführung« ausreichend sein. Wem dies dennoch zu wenig ist, muß in Zeile 1440 die betreffende Prüfung und in Zeile 60 die Dimensionierung anpassen.

Außerdem wäre das Scrollen des Bildschirms über eine Routine (ähnlich der in Zeile 5800-5830) zu steuern.

Buchen

Ein Aufruf dieser Funktion gibt zuerst die Nummern und Namen der Konten aus. Durch Eingabe der gewünschten Nummer wird das betreffende Konto eingelesen. Mit »E« kann man die Funktion ohne Aktivitäten verlassen. Anschließend wird nach Einnahme oder Ausgabe gefragt. An dieser Stelle kann die Routine ebenfalls durch die Taste »f7« abgebrochen werden. Diese Möglichkeiten wurden eingefügt, um ein Verlassen der Buchungsfunktion zu erlauben. Wurde jedoch dieser Punkt überschritten, bleibt nur, eine Buchung durchzuführen. Bei Abbruch werden die bis dahin getätigten Buchungen gesichert.

Nachdem Sie den Buchungsmonat angegeben haben, wird der alte Saldo angezeigt. Nun können »KOSTENART«, »BETRAG« und »TAG« (durch Kommata trennen) eingetippt werden. Wenn alles korrekt ist, können weitere Eingaben auf dem gewählten Konto gemacht werden. Wird die Frage nach weiteren Buchungen mit »N« beantwortet, werden die Eingaben auf das Konto »geschrieben«.

Nun können andere Konten aufgerufen oder die Funktion »Buchen« verlassen werden.

Kontenübersicht

Wählen Sie diese Funktion, wenn Sie die Monatssalden eines Kontos wünschen. Auch hier werden die Konten wieder vorgegeben.

Nach Anzeige der Übersicht ruft »f3« weitere ab oder »f1« druckt die aktuelle aus. Zum Menü kommt man mit »f5« zurück.

Monatsübersicht und Jahresübersicht

Tippen Sie die »5« im Menü und geben den gewünschten Monat ein, erhalten Sie einen Überblick über die Salden aller Konten für den betreffenden Monat.

Die »6« ruft den Jahressaldo aller Konten ab. Dabei kann allerdings der zu kumulierende Zeitraum (Monate) frei gewählt werden. Ober- und Untergrenze können in beliebiger Reihenfolge eingegeben werden.

Auch hier kann durch Drücken von »f1« ein Protokoll über Drucker erstellt werden.

Programm beenden

Mit dieser Funktion kann das Programm »ordnungsgemäß« beendet werden. Sollte dies mal aus Versehen geschehen sein, kann man mit »GOTO 960« sofort wieder zurückkehren. Es reicht, RETURN zu drücken, da der Cursor entsprechend positioniert wird.

Buchungsposten abfragen

Dieser Programmteil springt in das Untermenü »Sonderfunktion Buchungsposten bearbeiten«. Hier stehen als weitere Funktionen zur Verfügung:

<Datei laden>
2 Datei sichern
3 Buchungen ausgeben
4 Buchungen ändern
5 Buchungen sortieren
0 Sonderfunktionen verlassen

»Datei laden« braucht nicht gesondert aufgerufen zu werden, da dies wieder bei den übrigen Programmteilen des Untermenüs geschieht.

Vor dem Start der Laderoutine prüft der Computer, ob das Diskettenlaufwerk vorbereitet wurde. Auch eine »falsche« Datendisk wird mit »Datei nicht gefunden« quittiert. Es erfolgt Rücksprung zum Menü.

Datei sichern

Diese Funktion ist nur dann notwendig, wenn Änderungen (Kostenart oder Datum) durchgeführt wurden (vergleiche »Buchungen ändern«).

Allerdings kann ein unbeabsichtigtes Starten dieser Routine auch keinen »Schaden« anrichten. Es werden dann die Daten im Rechner gesichert und die identische Datei auf der Diskette gelöscht.

Darüber hinaus wird nach Aufruf dieses Programmteils zur Vorsicht gefragt, ob eine Diskette eingelegt wurde. Zusätzlich prüft der Computer, ob das Laufwerk »bereit« ist.

Buchungen ausgeben

Als erstes geben Sie ein, ob die Ausgabe auf Drucker oder Bildschirm erfolgen soll. Dann wählen Sie die Datei (mit dem betreffenden Kontennamen), aus der Buchungen selektiert werden sollen. Gesucht werden kann nach »Kostenart« und/oder »Buchungsmonat«. Ist ein Suchbegriff nicht relevant, drücken Sie RETURN. Wurde beide Male RETURN gedrückt, wird die Datei komplett aufgelistet. Nach jeweils 15 Buchungsposten wird die Bildschirmausgabe angehalten und mit Betätigen einer beliebigen Taste fortgesetzt.

Hier bieten sich also vielfältige Möglichkeiten, Kostenarten über Monate hinweg zu verfolgen oder Monate in die einzelnen Buchungspositionen zu »zerlegen«.

Buchungen ändern

Diese Funktion ist dazu bestimmt, Änderungen an den Buchungsposten vorzunehmen. Das gilt aber nur für Kostenart und Buchungsdatum, wenn man sich bei der Eingabe versehen und dies beim Buchen nicht bemerkt hat. Eine Änderung des Betrages ist dagegen nur durch eine Korrekturbuchung möglich, weil in diesem Fall auch die relativen Konten korrigiert werden müssen.

Gewählt werden kann zwischen zwei Änderungsmodi:
1. Selektion und Korrektur von Einzelbuchungen oder
2. Auflisten aller Buchungen

Bei Änderungen werden Spalten und Zeilen beziffert, so daß die Datensätze und die Felder eindeutig identifiziert werden können.

Nach Eingabe der betreffenden Zeilen- und Spaltennummer wird das zu ändernde Feld in die spezielle »Änderungszeile« ausgegeben. Der Cursor wird so positioniert, daß das (geänderte) Feld sofort wieder übernommen wird. Dies hat den Vorteil, daß bei nur einer Buchstabenänderung nicht das ganze Feld neu geschrieben werden muß.

Sind mehrere Änderungen vorzunehmen, können diese nacheinander durchgeführt werden.

Ist der letzte Datensatz erreicht, wird gefragt, ob die geänderte Datei gesichert werden soll. Dies wird in der Regel der Fall sein.

Buchungen sortieren

Die Buchungen werden in der Reihenfolge der Eingabe gespeichert. Wird für bestimmte Anwendungen eine andere Folge benötigt, kann nach Kostenart oder Betrag oder Datum aufsteigend sortiert werden.

Auch hier entscheidet der Anwender, ob die neu sortierte Datei gespeichert werden soll. In diesem Fall wird es nicht immer gewünscht sein, weil die Datei in einer bestimmten Reihenfolge auf der Diskette erhalten bleiben soll.

Ein Hinweis sei an dieser Stelle gegeben. Bei umfangreicheren Dateien (zum Beispiel 200 Datensätzen) können schon ein bis zwei Minuten vergehen, bis der Computer die Vollzugsmeldung gibt. Also nur keine Aufregung, wenn er sich nicht sofort wieder meldet. Es muß nicht unbedingt ein Programmfehler vorliegen!

Die 200 Datensätze (genauer 201 pro Konto) stellen die Obergrenze dar. »Vielbucher« müssen die Dimensionierung (D$(200,2)) in Zeile 60 ändern.

Sonderfunktion verlassen

Bevor das Programm zum Hauptmenü zurückspringt, prüft es, ob Änderungen oder Sortierungen an der sich im Computer befindenden Datei vorgenommen wurden. Sollten die noch nicht gesichert sein, kann dies jetzt — solange keine andere Datei von Diskette geladen wurde — nachgeholt werden. Andernfalls RETURN drücken, um zum Menü zu gelangen.

Aufschlüsselung nach Zeilennummern

Die folgende Grobeinteilung kann vorgenommen werden:

1 - 830	Vorbereitung und Unterprogramme, die der Schnelligkeit wegen, oder weil sie häufiger aufgerufen werden, an den Anfang gesetzt wurden (zum Beispiel Sortieren, Cursor positionieren).
850 - 4310	Hauptmenü und Bearbeitung der relativen Datei (Kontenführung).
4500 - 7780	Untermenü und Bearbeitung der sequentiellen Buchungsdateien.
7830 - 7940	DATAs für »INPUT«-Modifikation zum Einlesen von Files; werden in den Kassettenpuffer ab 828 geschrieben: Syntax: input# Filenummer, Satzlänge, Variable. Erlaubt mehr als 88 Zeichen (253) direkt einzulesen, findet Anwendung bei der relativen Datei, Satzlänge 141 Zeichen.
7990 - 8050	DATAs für die Simulation der »SWAP«-Funktion (vergleiche 64'er, Ausgabe 1/85) von Boris Schneider): Ermöglicht das Vertauschen zweier Strings ohne zusätzliche Variable (»String-Müll«), eingesetzt in der Sortierroutine. Syntax: SYS Startadresse(string1,string2) Startadresse im Programm: Speicherstelle 715 in der Zero-Page.

Wichtiger Hinweis:

Sollte jemand mit Simons Basic editieren, ist zu beachten, daß nach dem ersten Start des Programms nicht mehr alle Befehle dieser Basic-Erweiterung wegen der DATAs im Kassettenpuffer einwandfrei funktionieren (Syntax-Error!?)

Eine Beeinträchtigung von Exbasic Level II konnte bisher nicht beobachtet werden.

Gestartet wird das Programm mit RUN.

```
              Martin Schumann
==================================
H A U S H A L T S B U C H   1985
==================================

       FUNKTIONSAUSWAHL:
       -----------------

  -1-  KONTEN ANLEGEN
  -2-  BUCHEN

  -3-  KONTENUEBERSICHT

      <KONTENNAMEN AUSGEBEN>
        ----> 2-3,5-7

  -5-  MONATSUEBERSICHT
  -6-  JAHRESUEBERSICHT

  -7-  BUCHUNGSPOSTEN ABFRAGEN

  -0-  PROGRAMM BEENDEN

        AUSWAHL (0-7)

              Martin Schumann
==================================
H A U S H A L T S B U C H   1985
==================================

     SONDERFUNKTION
     -----------------
     BUCHUNGSPOSTEN BEARBEITEN

     <DATEI LADEN> ----> 3 - 5
  2  DATEI SICHERN

  3  BUCHUNGEN AUSGEBEN
  4  BUCHUNGEN AENDERN
  5  BUCHUNGEN SORTIEREN

  0  SONDERFUNKTION VERLASSEN

       AUSWAHL  0-5
```

Einzelne Menüs vom »Haushaltsbuch«

```
HAUS                    1984
KOSTENART                    BETRAG      DATUM
-----------------------------------------------

HAKEN/HOLZLEISTE              -5.90   09.10.84
BLUMENTOPF                    -4.90   10.10.84
ROLLTISCH/COMPUTER           -49.00   10.10.84
SESSEL/SCHLAFZ.             -164.00   13.10.84
CONCENTRA-BIRNEN              -6.00   29.10.84
TEPPICHWOLLE                -266.50   29.10.84
EICHEPANEELE                 -81.45   30.10.84
                         ----------
          SUMME             -577.75
```

Eine bestimmte Kostenart herausgegriffen

Variable	Bedeutung
A	= enthält bei Änderungen die Satznummer des zu ändernden Feldes
AK	= Anzahl der Konten (relative Datei)
AV	= Flag bei Änderungen
B	= Feldnummer des zu ändernden Feldes
BD$	= Buchungstag
BL$	= Leerstring mit 20 Positionen, wird beim Auffüllen der Inhalte von RC$ benutzt
C	= Zeilenposition des Cursors
D$(Y,I)	= Satz der sequentiellen Buchungsdatei Y,0 = Kostenart; Y,1 = Betrag Y,2 = Buchungsdatum
F%	= Normalwert in Speicherstelle 768 der Fehlerroutine des Interpreters
F$(I)	= Bezeichnung der Funktionen im Hauptmenü
FB$(I)	= Feldbezeichnung (sequentieller Datensatz)
FS$(I)	= Sonderfunktionen
FU	= gewählte Funktion des Untermenüs
GS	= Gesamtsaldo in den Übersichten
I,J,T,K,Y	= Laufvariable in Schleifen
J$	= aktuelles Jahr (4stellig)
JA$	= die beiden Endziffern des aktuellen Jahres
JS	= Jahressaldo aller Konten
KN$(I)	= Name der Konten (relative Datei)
KN	= Nummer des gewählten Kontos (Datei)
KZ	= Zahl der Konten (wird nur bei »Konten anlegen« benutzt; sonst AK)
L	= Position des Dezimalpunktes
LV	= Laufvariable (Buchen) für die Aufbereitung der Felder der sequentiellen Buchungsdateien
M$(I)	= Monatsnamen
MS(I)	= Jahressalden der Konten bei Jahresübersicht
MO$	= String für gewählten Monat (für Datum)
M	= Nummer des gewählten Monats
N$(I)	= Kurzform des Kontennamens für sequentielle Datei)
O	= Obergrenze bei Jahressaldo, wenn nicht das komplette Jahr kumuliert werden soll
RC$	= relativer Datensatz (Record)
U	= Untergrenze bei Jahressaldo, wenn nicht das komplette Jahr kumuliert werden soll.
S	= Spaltenposition des Cursors
S(M)	= Monatssalden des eingelesenen Kontos
S$(I)	= Monatssalden als String
S%	= Flag für Sortieren
SB$(I)	= enthält die Suchbegriffe auf
SU	= 1. bei »Buchungen«: Buchungsbetrag 2. bei »Buchungen ausgeben«: Summen der Ausgaben 3. übernimmt Salden für »Summen aufbereiten«
X	= gewählte Funktion
X$	= Eingaben in GET-Funktionen
Z	= Anzahl der Sätze einer sequentiellen Datei
ZS(M)	= Zwischenspeicher der Monatssalden der Konten
ZZ	= 1. Zwischenzähler in »Konten anlegen« für Zahl der tatsächlich eingegebenen Kontennamen 2. Zeilenzähler für Bildschirmausgabe
ZW	= Zwischenspeicher für die Nummer der aktuellen sequentiellen Datei im Computer

Tabelle 1. Die wichtigsten Variablen

Auf eines sei noch aufmerksam gemacht. Durch die Kommentarzeilen sind eine Vielzahl von Blöcken entstanden, die der Gliederung des Programms dienen, aber ohne Bedeutung für den Programmablauf sind. Es lassen sich einige Byte einsparen, wenn die Kommentare entfallen. Da sie keine Sprungadressen sind, kann dies auch bedenkenlos geschehen.

Das Programm selbst gewinnt jedenfalls dadurch an Schnelligkeit.

Nun wünsche ich allen Interessierten viel Erfolg beim Eingeben und Anwenden des »Haushaltsbuchs«. Möge die Kasse nun immer stimmen. (Martin Schumann/gk)

```
1 REM ***********************************            <250>
2 REM * > H A U S H A L T S B U C H < *              <046>
3 REM *                                *             <230>
4 REM *      MARTIN SCHUMANN           *             <015>
5 REM ***********************************            <254>
40 PRINT CHR$(147):POKE 211,12:POKE 214,10:SYS
   58732:PRINT"ICH LESE DATEN !"                     <224>
50 GOSUB 7830:GOSUB 7970:DIM SB$(2),F$(7),FS$(5
   )                                                 <184>
60 DIM MS(20),M$(12),D$(200,2),S(12),ZS(12),N$(
   20),S$(20),KN$(20):T1$="0"                        <071>
70 M$(1)="JANUAR{3SPACE}":M$(2)="FEBRUAR{2SPACE
   }"                                                <027>
80 M$(3)="MAERZ{4SPACE}":M$(4)="APRIL{4SPACE}"       <088>
90 M$(5)="MAI{6SPACE}":M$(6)="JUNI{5SPACE}"          <124>
100 M$(7)="JULI{5SPACE}":M$(8)="AUGUST{3SPACE}"      <138>
110 M$(9)="SEPTEMBER":M$(10)="OKTOBER{2SPACE}"       <111>
120 M$(11)="NOVEMBER ":M$(12)="DEZEMBER "            <147>
130 F$(1)=" KONTEN ANLEGEN{6SPACE}"                  <045>
140 F$(2)=" BUCHEN{14SPACE}"                         <036>
150 F$(3)=" KONTENUEBERSICHT{4SPACE}"                <055>
160 F$(4)=" KONTENNAMEN AUSGEBEN"                    <013>
170 F$(5)=" MONATSUEBERSICHT{4SPACE}"                <080>
180 F$(6)=" JAHRESUEBERSICHT{4SPACE}"                <070>
190 F$(7)=" BUCHUNGSPOSTEN ABFR."                    <039>
200 F$(0)=" PROGRAMM BEENDEN{4SPACE}"                <255>
210 F%=PEEK(768):GOTO 880                            <010>
220 REM                                              <107>
230 REM =*=  D R U C K E R  VORHANDEN?               <013>
240 REM                                              <127>
250 POKE 768,185:OPEN 1,4:PRINT#1,"":CLOSE 1:PO
    KE 768,F%                                        <245>
260 IF ST<>-128 GOTO 290                             <005>
270 POKE 214,24:POKE 211,2:SYS 58732:PRINT"{6SP
    ACE}DRUCKER NICHT BEREIT !{9SPACE}";             <039>
280 :FOR I=0 TO 2500:NEXT                            <038>
290 RETURN                                           <177>
300 REM                                              <188>
310 REM =*=   F L O P P Y   VORHANDEN?               <040>
320 REM                                              <208>
330 POKE 768,185:OPEN 1,8,15,"I":CLOSE 1:POKE 7
    68,F%                                            <034>
340 IF ST<>-128 GOTO 365                             <088>
350 C=23:S=9:GOSUB 830:PRINT"{RED,SPACE}FLOPPY
    NICHT BEREIT !"                                  <242>
360 :FOR I=0 TO 2500:NEXT:RETURN                     <062>
365 OPEN 1,8,15,"M-W"+CHR$(7)+CHR$(28)+CHR$(1)+
    CHR$(15):REM FLOPPYZUGRIFF BESCHL.               <084>
370 INPUT#1,R:CLOSE 1                                <177>
375 IF R=21 THEN S=9:C=23:GOSUB 830:PRINT"{RED}
    DISKETTE EINLEGEN !":FOR I=0 TO 3500:NEXT        <075>
380 RETURN                                           <011>
390 REM ==============================               <060>
400 REM **  SHELL - SORT /(NUM.)     **              <083>
410 REM ==============================               <080>
420 D=4                                              <207>
430 IF D<N THEN D=D+D:GOTO 430                       <168>
440 D=D-1                                            <207>
450 D=INT(D/2)                                       <226>
460 IF D<1 THEN RETURN                               <181>
470 FOR J=1 TO N-D                                   <102>
480 :FOR I=J TO 1 STEP-D                             <078>
490 : IF VAL(D$(I+D,X))>=VAL(D$(I,X))GOTO 540        <091>
500 :  FOR T=0 TO 2                                  <188>
510 :   SYS 715(D$(I,T),D$(I+D,T))                   <227>
520 :  NEXT T                                        <026>
530 :NEXT I                                          <025>
540 NEXT J                                           <234>
550 GOTO 450                                         <074>
570 REM ===========================                  <058>
580 REM **   SHELL - SORT/ALPHA      **              <255>
590 REM ===========================                  <078>
600 D=4                                              <132>
610 IF D<N THEN D=D+D:GOTO 610                       <093>
620 D=D-1                                            <132>
630 D=INT(D/2)                                       <151>
640 IF D<1 THEN RETURN                               <106>
650 FOR J=1 TO N-D                                   <027>
660 :FOR I=J TO 1 STEP-D                             <003>
670 : IF D$(I+D,X)>=D$(I,X)THEN 720                  <002>
680 :  FOR T=0 TO 2                                  <113>
690 :   SYS 715(D$(I,T),D$(I+D,T))                   <152>
700 : NEXT T                                         <206>
710 :NEXT I                                          <205>
720 NEXT J                                           <158>
730 GOTO 630                                         <254>
740 REM                                              <117>
750 REM *=* U E B E R S C H R I F T E N              <041>
760 PRINT TAB(8)"{ORANGE}UCCCCCCCCCCCCCCCCCCCCC
    CI":REM 22 CHR$(99)=SHIFT C                      <171>
770 PRINT TAB(8)"B";F$(X);" B":REM MENUEFUNKTIO
    NEN                                              <123>
780 PRINT TAB(8)"JCCCCCCCCCCCCCCCCCCCCCCK{BROWN
    }":REM COM-TASTE+2                               <143>
790 RETURN                                           <167>
810 REM =*= CURSOR POSITIONIEREN =*=                 <202>
830 POKE 211,S:POKE 214,C:SYS 58732:RETURN           <107>
850 REM **  ERSTER BILDSCHIRM      ****              <150>
860 REM ==============================               <081>
870 :                                                <163>
880 POKE 53281,7:POKE 53280,8:PRINT"{BROWN}":BL
    $="{20SPACE}":REM 20 BLANK                       <200>
890 GOSUB 4010:POKE 788,52:REM STOP AUS              <210>
900 INPUT"AKTUELLES JAHR (1984-1999): ";J$:JA$=
    RIGHT$(J$,2)                                     <179>
910 :IF J$<"1983"OR J$>"1999"THEN PRINT"UNZULAE
    SSIGES JAHR!":PRINT"{2UP}";:GOTO 900             <011>
920 REM        SHIFT/CRSR↑ <CHR$(145)>               <213>
930 REM ==============================               <151>
940 REM  MENUE / AUFRUF UNTERPROGRAMME               <232>
950 REM ==============================               <171>
960 GOSUB 4010                                       <021>
970 PRINT TAB(10)"{RED}FUNKTIONSAUSWAHL:"            <003>
980 PRINT TAB(10)"------------------":PRINT          <177>
990 PRINT TAB(5)"{BLUE}-1-{BROWN,2SPACE}KONTEN
    ANLEGEN"                                         <199>
1000 PRINT TAB(5)"{BLUE}-2-{BROWN,2SPACE}BUCHEN
     ":PRINT                                         <145>
1010 PRINT TAB(5)"{BLUE}-3-{BROWN,2SPACE}KONTEN
     UEBERSICHT":PRINT                               <164>
1020 PRINT TAB(5)"{2SPACE,ORANGE,2SPACE}<KONTEN
     NAMEN AUSGEBEN>":PRINT TAB(11)"----->{SPAC
     E,BLUE}2-3,5-7"                                 <058>
1030 PRINT TAB(45)"{BLUE}-5-{BROWN,2SPACE}MONAT
     SUEBERSICHT"                                    <031>
1040 PRINT TAB(5)"{BLUE}-6-{BROWN,2SPACE}JAHRES
     UEBERSICHT":PRINT                               <180>
1050 PRINT TAB(5)"{BLUE}-7-{BROWN,2SPACE}BUCHUN
     GSPOSTEN ABFRAGEN":PRINT                        <130>
1060 PRINT TAB(5)"{GREEN}-0-{BROWN,2SPACE}PROGR
     AMM BEENDEN":PRINT                              <108>
1070 PRINT TAB(10)"{RVSON,RED,SPACE}AUSWAHL (0-
     7){SPACE,RVOFF,BROWN}"                          <139>
1080 GET X$: IF X$<"0"OR X$>"7"THEN 1080             <137>
1090 : X=VAL(X$):IF X=0 GOTO 1200                    <103>
1100 : IF X<>4 GOTO 1130                             <083>
1110 :  S=6:C=22:GOSUB 830:PRINT"{RED}WERDEN BE
     I DEN UEBRIGEN FUNKTIO-"                        <001>
1120 :  PRINT TAB(6)"NEN AUTOMATISCH AUSGEGEBEN
     ";:FOR I=0 TO 2000:NEXT:GOTO 960                <027>
1130 : ON X GOSUB 1310,1750,2510,2890,3240,3580
     ,4500                                           <181>
1140 GOTO 960                                        <160>
1160 REM ==========================                  <077>
1170 REM ** PROGRAMM BEENDEN      **                 <035>
1180 REM ==========================                  <097>
1200 GOSUB 4010:GOSUB 760:PRINT:PRINT:PRINT          <227>
1210 CLOSE 1:CLOSE 2:CLOSE 3                         <168>
1220 PRINT"SOLLTEN SIE VERSEHENTLICH DAS PROGRA
     MM{2SPACE}BEENDET HABEN, SO KOENNEN ";          <036>
1230 PRINT"SIE ES MIT",TAB(15)CHR$(17)"{RVSON,R
     ED,SPACE}GOTO 960{SPACE,RVOFF,BROWN}"CHR$(
     17),,"WIEDER ";                                 <120>
1240 PRINT"STARTEN !":POKE 788,49:REM STOP EIN       <247>
1250 S=0:C=13:GOSUB 830:SYS 42115:REM END OHNE
     READY                                           <036>
1270 REM ***========================***              <195>
1280 REM ===  KONTEN ANLEGEN        ===              <203>
1290 REM ***========================***              <216>
1310 GOSUB 4010:GOSUB 760:REM GEW.FUNKT.             <105>
1320 PRINT"{BLUE}ACHTUNG ! EINE EVTL. BEREITS A
     NGELEGTE{2SPACE}DATEI DIESES JAHRES WIRD";      <202>
1330 PRINT" GELOESCHT.":PRINT CHR$(28)               <053>
1340 PRINT"{RED}WOLLEN SIE DAS (J/N)"                <220>
1350 GET X$:IF X$<>"J"AND X$<>"N"THEN 1350           <044>
1360 :IF X$="N"THEN RETURN                           <015>
1370 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN        <050>
1380 OPEN 3,8,15,"S:"+J$+"*":CLOSE 3:AK=0            <019>
1390 OPEN 2,8,15,"S:KONTEN"+J$                       <030>
1400 PRINT CHR$(147);:GOSUB 760                      <105>
1410 PRINT:PRINT" WIEVIELE KONTEN WOLLEN SIE AN
     LEGEN"                                          <012>
1420 PRINT"{2SPACE,RED,RVSON,SPACE}0{SPACE,RVOF
     F,BROWN,2SPACE}= KEINE KONTEN NEU ANLEGEN{
     3SPACE}":INPUT"(1-20) ";KZ                      <065>
1430 : IF KZ=0 THEN CLOSE 1:CLOSE 2:RETURN           <051>
1440 : IF KZ<1 OR KZ>20 THEN PRINT CHR$(145)CHR
     $(145)CHR$(145);:GOTO 1420                      <118>
1450 OPEN 1,8,2,"KONTEN"+J$+",L,"+CHR$(141)          <130>
1460 PRINT CHR$(145);:ZZ=0:D$(0,0)="NOCH KEINE
     BUCHUNGEN"                                      <183>
1470 S=1:C=4:GOSUB 830:PRINT"<{RED,RVSON}RETURN
     {RVOFF,BROWN}> = KONTEN ANLEGEN ABBRECHEN{
```

Listing 1. Das Haushaltsbuch. Beachten Sie bitte die Eingabehinweise auf Seite 77.

```
     2SPACE}"                                        <004>
1480 PRINT"{37SPACE}"                                <170>
1490 PRINT"{37SPACE}":REM 36 X SPACE                 <170>
1500 FOR I=1 TO KZ                                   <215>
1510 :PRINT"NAME KONTO NR.";SPC(3-LEN(STR$(I)))
     I":";                                           <216>
1520 : INPUT KN$                                     <113>
1530 :  IF KN$=""THEN I=KZ:GOTO 1610                 <073>
1540 :  IF LEN(KN$)>20 THEN KN$=LEFT$(KN$,20)        <045>
1550 : RC$=KN$+LEFT$(BL$,20-LEN(KN$))                <245>
1560 : N$(I)=LEFT$(RC$,12)                           <119>
1570 : FOR T=1 TO 12                                 <033>
1580 :  RC$=RC$+STR$(0)+LEFT$(BL$,10-LEN(STR$(0
     )))                                             <133>
1590 : NEXT T:ZZ=ZZ+1                                <123>
1600 : PRINT#1,RC$:KN$=""                            <027>
1610 NEXT I                                          <027>
1620 CLOSE 1:CLOSE 2:CLOSE 3                         <068>
1630 IF ZZ<KZ THEN KZ=ZZ:REM ABBRUCH ?               <172>
1640 FOR I=1 TO KZ                                   <100>
1650 : OPEN 3,8,3,J$+N$(I)+",S,W"                    <135>
1660 :  PRINT#3,D$(0,0)                              <248>
1670 : CLOSE 3                                       <153>
1680 NEXT I                                          <097>
1690 CLOSE 1:CLOSE 2:CLOSE 3:RETURN                  <082>
1710 REM ============================                <239>
1720 REM ***    B U C H E N        ***               <254>
1730 REM ============================                <003>
1750 GOSUB 2900:IF KN$="E"OR AK=0 THEN RETURN        <068>
1760 IF ST=-128 OR R=21 THEN RETURN                  <092>
1770 X=2:GOSUB 4110:LV=1                             <120>
1780 PRINT CHR$(147)"{LIG.BLUE}";:GOSUB 760:SU=
     0                                               <136>
1790 PRINT"--------------------------------"         <084>
1800 PRINT"{LIG.BLUE}NR.";KN;" -{SPACE,RED}";KN
     $(KN)                                           <210>
1810 PRINT"{BROWN}------------------------------
     --"                                             <254>
1820 PRINT"EINNAHME ODER AUSGABE (E/A) ?"            <172>
1830 PRINT"--------------------------------"         <125>
1840 PRINT                                           <208>
1850 PRINT"{BLUE,SPACE}FUNKTION VERLASSEN MIT{S
     PACE,RED}F7{BROWN}!";                           <210>
1860 GET X$:IF X$<>"E"AND X$<>"A"THEN IF X$<>CH
     R$(136)GOTO 1860                                <198>
1870 : IF X$=CHR$(136)AND LV>1 THEN LV=LV-1:GOT
     O 2210                                          <046>
1880 : IF X$=CHR$(136)THEN CLOSE 1:CLOSE 2:RETU
     RN                                              <080>
1890 BA$=X$:REM BUCHUNGSART MERKEN                   <015>
1900 S=30:C=6:GOSUB 830:PRINT BA$:S=1:C=9:GOSUB
      830                                            <086>
1910 INPUT"MONAT (1-12){7SPACE}:{5SPACE,BLUE}";
     MO$:M=VAL(MO$)                                  <058>
1920 : IF M<1 OR M>12 THEN PRINT CHR$(145);:GOT
     O 1910                                          <073>
1930 PRINT"{BROWN}------------------------------
     --"                                             <118>
1940 PRINT"ALTER STAND{6SPACE}:{SPACE,LIG.BLUE}
     ";S(M):ZS(M)=S(M)                               <146>
1950 PRINT"{BROWN}------------------------------
     --"                                             <138>
1960 PRINT"KOSTENART, BETRAG, TAG{2SPACE}:{BLUE
     }"                                              <137>
1970 INPUT D$(LV,0),SU,BD$                           <159>
1980 PRINT"{BROWN}------------------------------
     --"                                             <168>
1990 IF X$="E"THEN S(M)=S(M)+SU:GOTO 2010            <034>
2000 S(M)=S(M)-SU                                    <190>
2010 PRINT"NEUER STAND{6SPACE}:{SPACE,BLUE}";S(
     M)                                              <060>
2020 PRINT"{BROWN}------------------------------
     --":PRINT                                       <163>
2030 PRINT"{LIG.BLUE}ALLES RICHTIG (J/N) ?{BROW
     N}"                                             <211>
2040 GET X$:IF X$<>"J"AND X$<>"N"THEN 2040           <221>
2050 IF X$="N"THEN S(M)=ZS(M):GOTO 1780              <125>
2090 IF LEN(BD$)=1 THEN BD$=T1$+BD$                  <094>
2110 :::REM *=*= TAG ZU DATUM AUFBEREITEN
      *=*= MONAT/TAG/JAHR                            <045>
2120 : D$(LV,2)=MO$+BD$+JA$                          <098>
2130 : IF BA$="A"THEN SU=SU-2*SU                     <215>
2140 : GOSUB 4370:REM NACHKOMMASTELLE PRUEFEN        <047>
2150 : IF LEN(D$(LV,0))>20 THEN D$(LV,0)=LEFT$(
     D$(LV,0),20)                                    <127>
2160 : D$(LV,1)=SU$                                  <232>
2180 PRINT:PRINT"{BLUE}WEITERE BUCHUNGEN AUF DI
     ESEM KONTO{6SPACE}(J/N) ?{BROWN}"               <025>
2190 GET X$:IF X$<>"J"AND X$<>"N"THEN 2190           <122>
2200 :IF X$="J"THEN LV=LV+1:SU=0:PRINT CHR$(147
     );:GOSUB 760:GOTO 1790                          <023>
2210 PRINT CHR$(147):POKE 211,10:POKE 214,11:SY
     S 58732                                         <192>
2220 PRINT"{LIG.BLUE}OK. ICH SPEICHERE !"            <128>
2230 RC$=""                                          <109>
2240 RC$=RC$+KN$+LEFT$(BL$,20-LEN(KN$))              <210>
2250 FOR I=1 TO 12                                   <134>
2260 : S$=STR$(S(I))                                 <065>
2270 : RC$=RC$+S$+LEFT$(BL$,10-LEN(S$))              <157>
2280 NEXT I                                          <187>
2290 PRINT#2,"P"+CHR$(2)+CHR$(KN)+CHR$(0)+CHR$(
     1)                                              <184>
2300 PRINT#1,RC$                                     <178>
2310 CLOSE 1:CLOSE 2:CLOSE 3                         <249>
2320 :                                               <083>
2330 REM == ABSPEICHERN DER SEQ.DATEI                <169>
2340 REM == DATEN ANHAENGEN ← "APPEND"               <130>
2350 N$=LEFT$(KN$(KN),12)                            <170>
2360 OPEN 3,8,3,J$+N$+",S,A"                         <102>
2370 FOR Y=1 TO LV                                   <078>
2380 : FOR I=0 TO 2                                  <017>
2390 :  PRINT#3,D$(Y,I)                              <023>
2400 : NEXT I                                        <110>
2410 NEXT Y                                          <078>
2420 CLOSE 3                                         <080>
2430 PRINT CHR$(145);:PRINT"{RED}WEITERE BUCHUN
     GEN AUF ANDEREN KONTEN{4SPACE}(J/N) ?{BROW
     N}"                                             <136>
2440 GET X$:IF X$<>"J"AND X$<>"N"THEN 2440           <115>
2450 IF X$="J" THEN 1750                             <086>
2460 RETURN                                          <051>
2470 REM ============================                <234>
2480 REM ***  KONTEN-UEBERSICHT  ****                <088>
2490 REM ============================                <254>
2500 :                                               <007>
2510 GOSUB 2900:IF KN$="E"OR AK=0 THEN RETURN        <063>
2520 X=3:GOSUB 4110:IF ST=-128 OR R=21 THEN RET
     URN                                             <091>
2530 PRINT CHR$(147);:GOSUB 760                      <215>
2540 PRINT"{LIG.BLUE}---------------------------
     ------------{BLUE}"                             <057>
2550 PRINT" NR.";KN;"- ";"{RED}"KN$"{LIG.BLUE}"      <096>
2560 PRINT"-------------------------------------
     --"                                             <149>
2570 PRINT SPC(2)"MONAT"SPC(22)"SALDO"               <091>
2580 PRINT"-------------------------------------
     --{BROWN}"                                      <062>
2590 GS=0                                            <164>
2600 FOR I=1 TO 12                                   <230>
2610 : PRINT SPC(2)M$(I);SPC(22-LEN(STR$(INT(S(
     I)))))"{LIG.BLUE}"S(I)"{BROWN}"                 <244>
2620 : GS=GS+S(I)                                    <253>
2630 NEXT I                                          <027>
2640 PRINT"-------------------------------------
     --"                                             <229>
2650 PRINT SPC(2)"{RED}"J$;SPC(3)"{BROWN}GESAMT
     "SPC(18-LEN(STR$(INT(GS))))"{BLUE}"GS           <076>
2660 PRINT SPC(27)"=========="                      <229>
2670 CLOSE 1:CLOSE 2:IF X$=CHR$(133)THEN PRINT#
     4:CLOSE 4:GOTO 2780                             <012>
2680 PRINT"WEITERE UEBERSICHTEN ={SPACE,RED}F3{
     BLUE}"                                          <230>
2690 GOTO 2760                                       <228>
2700 RETURN                                          <036>
2720 REM ============================                <229>
2730 REM =*= UP AUSDRUCKEN ?      ===                <119>
2740 REM ============================                <249>
2760 POKE 214,24:POKE 211,0:SYS 58732                <052>
2770 PRINT"{2SPACE}AUSDRUCKEN ={SPACE,RED}F1{BL
     UE,3SPACE}ZUM MENUE{2SPACE}={SPACE,RED}F5{
     BLUE}";:PRINT CHR$(145);                        <001>
2780 : GET X$:IF X$=CHR$(133)GOTO 2820               <052>
2790 :  IF X$=CHR$(134)AND X=3 GOTO 2510             <208>
2800 :IF X$=CHR$(135)GOTO 960                        <198>
2810 : GOTO 2780                                     <152>
2820 GOSUB 250:IF ST=-128 THEN X$="":GOTO 2760       <066>
2830 CLOSE 4:OPEN 4,4:PRINT#4,CHR$(27)+CHR$(108
     )+CHR$(10):CMD 4:IF X=3 GOTO 2540               <236>
2840 :  IF X=5 GOTO 3420                             <121>
2850 :  IF X=6 GOTO 3850                             <139>
2860 :                                               <113>
2870 REM ============================                <124>
```

Listing 1. Das Haushaltsbuch (Fortsetzung)

```
2880 REM ** KONTENNAMEN AUSGEBEN  ***                  <052>
2890 REM ============================                  <144>
2900 X=4:PRINT CHR$(147);:GOSUB 760:PRINT              <150>
2910 IF AK<>0 THEN GOSUB 3120:RETURN                   <214>
2920 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN          <070>
2930 OPEN 1,8,2,"KONTEN"+J$+",L,"+CHR$(141)            <080>
2940 OPEN 2,8,15                                       <078>
2950 : AK=1                                            <058>
2960 : PRINT#2,"P"+CHR$(2)+CHR$(AK)+CHR$(0)+CHR
     $(1)                                              <134>
2970 : RC$=""                                          <142>
2980 : INPUT*1,141,RC$:IF LEFT$(RC$,1)="£"THEN
     RC$="":GOTO 3040                                  <187>
2990 :  INPUT#2,R                                      <039>
3000 :  IF R=50 THEN RC$="":GOTO 3040                  <209>
3010 : KN$(AK)=LEFT$(RC$,20):L%=LEN(STR$(AK))          <197>
3020 : PRINT SPC(3-L%)AK;" - ";KN$(AK)                 <213>
3030 : AK=AK+1:GOTO 2960                               <084>
3040 AK=AK-1:CLOSE 1:CLOSE 2:IF AK<>0 GOTO 3060 <223>
3050 PRINT TAB(6)"{RED}DATEI "J$" NICHT GEFUNDE
     N !{BROWN}":FOR I=0 TO 2000:NEXT:RETURN           <081>
3060 GOSUB 3130:RETURN                                 <027>
3070 :                                                 <067>
3080 REM ** K O N T O  AUSWAEHLEN  **                  <195>
3090 REM ============================                  <089>
3110 GOSUB 4990                                        <149>
3120 FOR I=1 TO AK:L%=LEN(STR$(I)):PRINT SPC(3-
     L%)I;KN$(I):NEXT I                                <185>
3130 PRINT:PRINT"ABBRECHEN MIT{SPACE,RVSON,RED,
     SPACE}E{SPACE,RVOFF,BROWN}"                       <244>
3140 INPUT"KONTONUMMER (1-20) : ";KN$                  <187>
3150 : IF KN$="E"OR AK=0 THEN RETURN                   <106>
3160 : KN=VAL(KN$)                                     <188>
3170 : IF KN<1 OR KN>AK THEN PRINT CHR$(145);:G
     OTO 3140                                          <238>
3180 RETURN                                            <006>
3200 REM ============================                  <199>
3210 REM ** MONATSUEBERSICHT       ***                 <183>
3220 REM ============================                  <219>
3240 PRINT CHR$(147);:GOSUB 760:PRINT                  <115>
3250 INPUT"MONAT{2SPACE}: ";M                          <200>
3260 : IF M<1 OR M>12 THEN PRINT CHR$(145);:GOT
     O 3250                                            <137>
3270 :                                                 <012>
3280 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN          <175>
3290 OPEN 1,8,2,"KONTEN"+J$+",L,"+CHR$(141)            <185>
3300 OPEN 2,8,15                                       <183>
3310 GS=0:KN=1                                         <044>
3320 PRINT#2,"P"+CHR$(2)+CHR$(KN)+CHR$(0)+CHR$(
     1)                                                <194>
3330 :RC$=""                                           <248>
3340 : INPUT*1,141,RC$:IF LEFT$(RC$,1)="£"THEN
     RC$="":GOTO 3410                                  <038>
3350 : INPUT#2,R                                       <145>
3360 :  IF R=50 THEN RC$="":GOTO 3410                  <028>
3370 : KN$(KN)=LEFT$(RC$,20)                           <042>
3380 : S$(KN)=MID$(RC$,(20+(M-1)*10),10)               <255>
3390 : GS=GS+VAL(S$(KN))                               <140>
3400 : KN=KN+1:GOTO 3320                               <217>
3410 SU=GS:GOSUB 4370                                  <232>
3420 PRINT CHR$(145)"----------{SPACE,RED}";M$(
     M);" "J$"{SPACE,BROWN}---------"                  <078>
3430 PRINT"{BLUE}NR.{3SPACE}NAME{16SPACE}SALDO" <209>
3440 PRINT"{BROWN}------------------------------
     -----"                                            <233>
3450 FOR K=1 TO KN-1:L%=LEN(STR$(K))                   <032>
3460 : PRINT SPC(3-L%)K;KN$(K);SPC(6-LEN(STR$(I
     NT(VAL(S$(K)))))"{LIG.BLUE}"S$(K)"{BROWN}
     "                                                 <110>
3470 NEXT                                              <029>
3480 PRINT"{BROWN}------------------------------
     -----"                                            <017>
3490 PRINT M$(M)" "J$;SPC(3)"GESAMT "SPC(7-(L-1
     ))"{BLUE}"SU$"{BROWN}"                            <125>
3500 PRINT SPC(24)"=========="                         <045>
3510 CLOSE 1:CLOSE 2:IF X$=CHR$(133)THEN PRINT#
     4:CLOSE 4                                         <195>
3520 GOTO 2760                                         <037>
3540 REM *** ====================== **                 <128>
3550 REM ***   JAHRESUEBERSICHT     **                 <247>
3560 REM *** ====================== **                 <148>
3580 PRINT CHR$(147);:GOSUB 760:PRINT                  <200>
3590 PRINT"{2SPACE}SALDO{2SPACE}12{2SPACE}MONAT
     E{2SPACE}( J/ N ) ?":PRINT:PRINT                  <136>
3600 GET X$:IF X$<>"J"AND X$<>"N"GOTO 3600             <225>
3610 : IF X$="J"THEN U=1:O=12:GOTO 3660                <184>
3620 PRINT"GEBEN SIE BITTE UNTER- UND OBERGRENZ
     E AN":INPUT"(U,O) --> ";U,O                       <180>
3630 IF U<1 OR U>12 OR O<1 OR O>12 THEN PRINT C
     HR$(145)CHR$(145)CHR$(145);:GOTO 3620             <046>
3640 IF U>O THEN T=U:U=O:O=T                           <071>
3660 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN          <045>
3670 PRINT CHR$(147);:GOSUB 760:PRINT                  <035>
3680 OPEN 1,8,2,"KONTEN"+J$+",L,"+CHR$(141)            <065>
3690 OPEN 2,8,15                                       <063>
3700 JS=0:FOR K=1 TO AK:MS(K)=0:NEXT:KN=1              <228>
3710 : PRINT#2,"P"+CHR$(2)+CHR$(KN)+CHR$(0)+CHR
     $(1)                                              <132>
3720 ::INPUT*1,141,RC$:IF LEFT$(RC$,1)="£"THEN
     RC$="":GOTO 3840                                  <228>
3730 :  INPUT#2,R                                      <014>
3740 :  IF R=50 THEN RC$="":GOTO 3840                  <192>
3750 : KN$(KN)=LEFT$(RC$,20)                           <167>
3760 : FOR M=U TO O                                    <192>
3770 :  S$(KN)=" ":S=0                                 <200>
3780 :  S$(KN)=MID$(RC$,(20+(M-1)*10),10)              <144>
3790 :  S=VAL(S$(KN))                                  <146>
3800 :  MS(KN)=MS(KN)+S                                <227>
3810 : NEXT M                                          <249>
3820 : JS=JS+MS(KN)                                    <084>
3830 KN=KN+1:GOTO 3710                                 <081>
3840 SU=JS:GOSUB 4370                                  <155>
3850 PRINT"{BROWN}--------------{SPACE,RED}"J$"
     {BROWN,SPACE}---------------"                     <218>
3860 PRINT"{BLUE}NR.{2SPACE}NAME{17SPACE}SALDO" <129>
3870 PRINT"{BROWN}------------------------------
     -----"                                            <153>
3880 FOR K=1 TO KN-1:L%=LEN(STR$(K))                   <208>
3890 :PRINT SPC(3-L%)K;KN$(K);SPC(6-LEN(STR$(IN
     T(MS(K)))))"{LIG.BLUE}"MS(K)"{BROWN}"             <090>
3900 NEXT                                              <205>
3910 PRINT"{BROWN}------------------------------
     -----"                                            <193>
3920 PRINT"{SPACE,RED}";J$;" SALDO ";                  <046>
3930 PRINT LEFT$(M$(U),3)" - "LEFT$(M$(O),3)SPC
     (9-(L-1))"{BLUE}"SU$"{BROWN}"                     <191>
3940 PRINT SPC(23)"==========="                        <035>
3950 CLOSE 1:CLOSE 2:IF X$=CHR$(133)THEN PRINT#
     4:CLOSE 4                                         <125>
3960 GOTO 2760                                         <223>
3980 REM ========================== **                 <176>
3990 REM *** PROGRAMMKOPF ERSTELLEN **                 <073>
4000 REM ========================== **                 <196>
4010 PRINT CHR$(147);                                  <065>
4020 PRINT TAB(4)"==============================
     ==="                                              <064>
4030 PRINT TAB(4)"H A U S H A L T S B U C H{3SP
     ACE}"+J$                                          <145>
4040 PRINT TAB(4)"==============================
     ==="                                              <084>
4050 PRINT:RETURN                                      <066>
4070 REM *** ====================== **                 <148>
4080 REM **    KONTO EINLESEN       **                 <020>
4090 REM *** ====================== **                 <168>
4110 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN          <241>
4120 PRINT CHR$(147);:GOSUB 760:PRINT:PRINT            <186>
4130 PRINT:PRINT"{LIG.BLUE}KONTO{SPACE,RED}";KN
     $(KN);"{LIG.BLUE,SPACE}WIRD EINGELE-SEN !{
     BROWN}"                                           <022>
4140 OPEN 1,8,2,"KONTEN"+J$+",L,"+CHR$(141)            <015>
4150 OPEN 2,8,15                                       <013>
4160 PRINT#2,"P"+CHR$(2)+CHR$(KN)+CHR$(0)+CHR$(
     1)                                                <014>
4170 RC$=""                                            <009>
4180 :INPUT*1,141,RC$:IF LEFT$(RC$,1)="£"THEN R
     C$="":GOTO 4310                                   <113>
4190 : INPUT#2,R                                       <220>
4200 :  IF R<>50 THEN 4250                             <202>
4210 PRINT:PRINT"JAHRESDATEI BZW. KONTO NICHT G
     EFUNDEN !":PRINT                                  <184>
4220 PRINT"WEITER MIT{SPACE,RVOFF,LIG.GREEN,SPA
     CE}TASTE{SPACE,RVOFF,BROWN,SPACE}!";:WAIT
     198,1:POKE 198,0                                  <236>
4230 CLOSE 1:CLOSE 2:RETURN                            <059>
4240 :                                                 <218>
4250 KN$=LEFT$(RC$,20)                                 <121>
4260 GS=0                                              <048>
4270 FOR I=1 TO 12                                     <114>
4280 : S(I)=VAL(MID$(RC$,20+(I-1)*10,10))              <211>
4290 : GS=GS+S(I)                                      <137>
4300 NEXT I                                            <167>
4310 RETURN                                            <116>
```

Listing 1. Das Haushaltsbuch (Fortsetzung)

```
4320 :                                             <042>
4330 REM ==============================           <175>
4340 REM = UP SUMMENAUSGABE AUFBEREITEN            <108>
4350 REM =    <PRINT USING>-SIMULATION             <153>
4360 REM ==============================           <206>
4370 SU$=STR$(SU):A=LEN(SU$):L=0:FOR T=1 TO A:I
     F MID$(SU$,T,1)="."THEN L=T:T=A               <209>
4380 NEXT T                                        <003>
4390 : IF L=0 THEN SU$=SU$+".00":L=A+1:RETURN      <179>
4400 : IF A>=L+2 AND SU<0 THEN SU=SU-.00001:SU$
     =LEFT$(STR$(SU),L+2):RETURN                   <041>
4410 : IF A>=L+2 AND SU>0 THEN SU=SU+.00001:SU$
     =LEFT$(STR$(SU),L+2):RETURN                   <048>
4420 : IF A=L+1 THEN SU$=SU$+"0"                   <067>
4430 RETURN                                        <237>
4440 :                                             <163>
4450 REM ============================             <174>
4460 REM ==   SONDERFUNKTIONEN     ==              <204>
4470 REM ==  SEQ.BUCHUNGSDATEIEN   ==              <122>
4480 REM ============================             <204>
4490 :                                             <213>
4500 FS$(1)="DATEI LADEN{7SPACE,SHIFT-SPACE}"      <069>
4510 FS$(2)="DATEI SICHERN{5SPACE,SHIFT-SPACE}"    <248>
4520 FS$(3)="BUCHUNGEN AUSGEBEN{SHIFT-SPACE}"      <121>
4530 FS$(4)="BUCHUNGEN AENDERN{SPACE,SHIFT-SPAC
     E}"                                           <055>
4540 FS$(5)="BUCHUNGEN SORTIEREN"                  <096>
4550 FS$(0)="S-FUNKTION BEENDEN "                  <234>
4560 :                                             <027>
4570 FB$(0)="KOSTENART{4SPACE}"                    <201>
4580 FB$(1)="BETRAG{7SPACE}"                       <206>
4590 FB$(2)="BUCHUNGSMONAT"                        <002>
4600 :S%=0                                         <157>
4610 GOSUB 3990:PRINT                              <073>
4620 PRINT TAB(8)"{RED}SONDERFUNKTION":PRINT TA
     B(8)"----------------"                        <108>
4630 PRINT TAB(8)"{BLUE}BUCHUNGSPOSTEN BEARBEIT
     EN":PRINT                                     <004>
4640 PRINT TAB(5)"{2SPACE,ORANGE,SPACE}<DATEI L
     ADEN> ---->{SPACE,RVSON,BLUE}3{RVOFF,SPACE
     }-{SPACE,RVSON}5{RVOFF}"                      <164>
4650 PRINT TAB(5)"{BLUE,RVSON}2{RVOFF,BROWN,2SP
     ACE}DATEI SICHERN":PRINT                      <234>
4660 PRINT TAB(5)"{BLUE,RVSON}3{RVOFF,BROWN,2SP
     ACE}BUCHUNGEN AUSGEBEN"                       <152>
4670 PRINT TAB(5)"{BLUE,RVSON}4{RVOFF,BROWN,2SP
     ACE}BUCHUNGEN AENDERN"                        <086>
4680 PRINT TAB(5)"{BLUE,RVSON}5{RVOFF,BROWN,2SP
     ACE}BUCHUNGEN SORTIEREN":PRINT:PRINT          <197>
4690 PRINT TAB(5)"{GREEN}0{BROWN,2SPACE}SONDERF
     UNKTION VERLASSEN":PRINT:PRINT                <183>
4700 :                                             <168>
4710 PRINT TAB(8)"{RED,RVSON,SPACE}AUSWAHL{2SPA
     CE}0-5{SPACE,RVOFF,BROWN,SPACE}"              <085>
4720 :                                             <188>
4730 GET F$:IF F$="0"THEN CLOSE 3:GOSUB 4820:RE
     TURN:REM ZUM HAUPTMENUE                       <132>
4740 IF F$<"2"OR F$>"5"THEN 4730                   <109>
4750 FU=VAL(F$)                                    <109>
4760 :ON FU GOSUB 6300,6560,5070,6880,6030         <082>
4770 GOTO 4610                                     <008>
4780 :                                             <248>
4790 REM =*=*=  SONDERFUNKTIONEN VERLASSEN *=*     <114>
4800 REM PRUEFEN, OB AENDERUNGEN GESICHERT =*=     <161>
4810 :                                             <022>
4820 IF AV=0 AND S%=0 THEN RETURN                  <040>
4830 PRINT CHR$(147):GOSUB 4990:PRINT              <173>
4840 IF AV=0 AND S%=1 GOTO 4890                    <102>
4850 PRINT"DIE DATEI{SPACE,RED}"KN$(ZW)"{BROWN,
     SPACE}WURDE","VERAENDERT, ABER NICHT GESIC
     HERT !"                                       <048>
4860 PRINT:PRINT"WOLLEN SIE SICHERN ? --->{2SPA
     CE,RED,RVSON,SPACE}S{SPACE,RVOFF,BLUE}"       <178>
4870 PRINT:PRINT"ZURUECK ZUM MENUE{2SPACE}? ---
     >{SPACE,RED}RETURN{BLUE}":GOSUB 4920:AV=0:
     RETURN                                        <152>
4880 :                                             <093>
4890 PRINT"DATEI{SPACE,RED}"KN$(ZW)"{BROWN,SPAC
     E}WURDE","SORTIERT, ABER NOCH NICHT GESICH
     ERT !"                                        <059>
4900 PRINT:PRINT"WOLLEN SIE SICHERN ? --->{2SPA
     CE,RED,RVSON,SPACE}S{SPACE,RVOFF,BLUE}":PR
     INT                                           <174>
4910 PRINT"ZURUECK ZUM MENUE{2SPACE}? --->{SPAC
     E,RED}RETURN{BLUE}"                           <220>
4920 GET X$:IF X$=CHR$(13)THEN RETURN              <012>
4930 IF X$="S"THEN S%=0:GOSUB 6590:RETURN          <009>
4940 GOTO 4920                                     <183>
4950 :                                             <163>
4960 REM =*= UEBERSCHRIFT / FUNKTION               <201>
4970 REM ============================             <184>
4980 :                                             <193>
4990 PRINT TAB(8)"{ORANGE}UCCCCCCCCCCCCCCCCCCC
     CCI"                                          <083>
5000 PRINT TAB(8)"B ";FS$(FU);"{2SPACE}B"          <099>
5010 PRINT TAB(8)"JCCCCCCCCCCCCCCCCCCCCCK{BROW
     N}"                                           <114>
5020 PRINT:RETURN                                  <016>
5030 :                                             <243>
5040 REM ============================             <254>
5050 REM == BUCHUNGEN AUSGEBEN      ==             <057>
5060 REM ============================             <018>
5070 PRINT CHR$(147);                              <105>
5080 GOSUB 4990:PRINT:PRINT"{BLUE}DRUCKER ({RVS
     ON,RED}D{RVOFF,BLUE}) ODER BILDSCHIRM ({RV
     SON,RED}B{RVOFF,BLUE}) ?"                     <151>
5090 GET A$:AA$=A$:IF A$<>"D"AND A$<>"B"GOTO 50
     90                                            <093>
5100 GOSUB 6300:IF KN$="E"OR AK=0 THEN RETURN      <101>
5110 IF ST=-128 OR R=21 THEN RETURN                <127>
5120 IF A$<>"D"GOTO 5180                           <071>
5130 GOSUB 250:IF ST=-128 THEN ZW=0:RETURN         <168>
5140 CLOSE 4:OPEN 4,4:PRINT#4,CHR$(27)+CHR$(108
     )+CHR$(10);                                   <213>
5145 PRINT#4,CHR$(10);CHR$(14);KN$(KN);"{SHIFT-
     SPACE}";J$;CHR$(10)                           <180>
5150 PRINT#4,"KOSTENART{18SPACE}BETRAG{7SPACE}D
     ATUM{3SPACE}"                                 <089>
5160 PRINT#4,"-----------------------------------
     ---------------":PRINT#4                      <238>
5170 IF Z=0 THEN PRINT#4:PRINT#4,SPC(9)D$(Z,0):
     PRINT#4:CLOSE 4:RETURN                        <005>
5180 FU=3:SU=0:ZZ=0                                <100>
5190 PRINT CHR$(147);:GOSUB 4990:PRINT:PRINT KN
     $(KN):PRINT                                   <159>
5200 IF Z=0 THEN PRINT TAB(90)"{RED}"D$(Z,0):FO
     R I=0 TO 2000:NEXT:RETURN                     <063>
5210 : SB$(0)="":SB$(2)=""                         <068>
5220 POKE 211,3:POKE 214,13:SYS 58732:PRINT"{BL
     UE}DRUECKEN SIE BEI NICHT RELEVANTEN"         <125>
5230 PRINT TAB(3)"FELDERN{2SPACE}---->{2SPACE,R
     ED}RETURN{BROWN}"                             <225>
5240 POKE 214,7:PRINT                              <240>
5250 : PRINT FB$(0);:INPUT SB$(0)                  <202>
5260 : PRINT FB$(2);:INPUT SB$(2)                  <216>
5270 IF SB$(2)<>""AND VAL(SB$(2))<1 OR VAL(SB$(
     2))>12 THEN PRINT CHR$(145);:GOTO 5260        <209>
5280 PRINT CHR$(147);:GOSUB 4990:PRINT"{RED}"KN
     $(KN)" "J$:PRINT"{BROWN}":IF FU=4 GOTO 530
     0                                             <168>
5290 IF SB$(0)=""AND SB$(2)=""AND A$<>"D"THEN F
     OR Y=1 TO Z:GOSUB 5680:NEXT Y:GOTO 5470       <254>
5300 FOR Y=1 TO Z                                  <131>
5310 : SB=0                                        <131>
5320 : IF SB$(0)="" THEN SB=1:GOTO 5340            <127>
5330 : IF LEFT$(D$(Y,0),LEN(SB$(0)))=SB$(0) THE
     N SB=SB+1                                     <246>
5340 : IF SB$(2)=""THEN SB=SB+1:GOTO 5380          <216>
5350 :::REM =*= DATUM 1- OD. 2-STELLIG?            <039>
5360 : IF LEN(D$(Y,2))=5 AND LEFT$(D$(Y,2),1)=S
     B$(2)THEN SB=SB+1:GOTO 5380                   <168>
5370 : IF LEN(D$(Y,2))=6 AND LEFT$(D$(Y,2),2)=S
     B$(2)THEN SB=SB+1                             <033>
5380 :IF FU=4 GOTO 5420                            <179>
5390 :IF A$<>"D"GOTO 5410                          <140>
5400 : IF SB=2 AND FU=3 THEN GOSUB 5860:GOTO 54
     40                                            <049>
5410 : IF SB=2 AND FU=3 THEN GOSUB 5680:GOTO 54
     40                                            <059>
5420 : IF SB=2 AND ZZ>0 THEN GOSUB 7150:GOTO 54
     40                                            <084>
5430 : IF SB=2 AND ZZ<1 THEN GOSUB 7140            <208>
5440 NEXT Y                                        <048>
5450 IF FU=4 OR E$="E"THEN RETURN:REM ZU "AENDE
     RUNGEN"                                       <220>
5460 IF A$="D"THEN GOSUB 5960:GOTO 5500            <040>
5470 PRINT TAB(20)"----------":PRINT TAB(10)"{L
     IG.BLUE}SUMME{BLUE,SPACE}";:GOSUB 4370        <148>
5480 PRINT TAB(26-(L-1))SU$;"{BROWN}"              <078>
5490 ZZ=0                                          <029>
5500 PRINT:PRINT"{RED}DATEIENDE ERREICHT."         <100>
5510 PRINT"SUCHBEGRIFF NICHT (NOCHMALS) GEFUNDE
```

Listing 1. Das Haushaltsbuch (Fortsetzung)

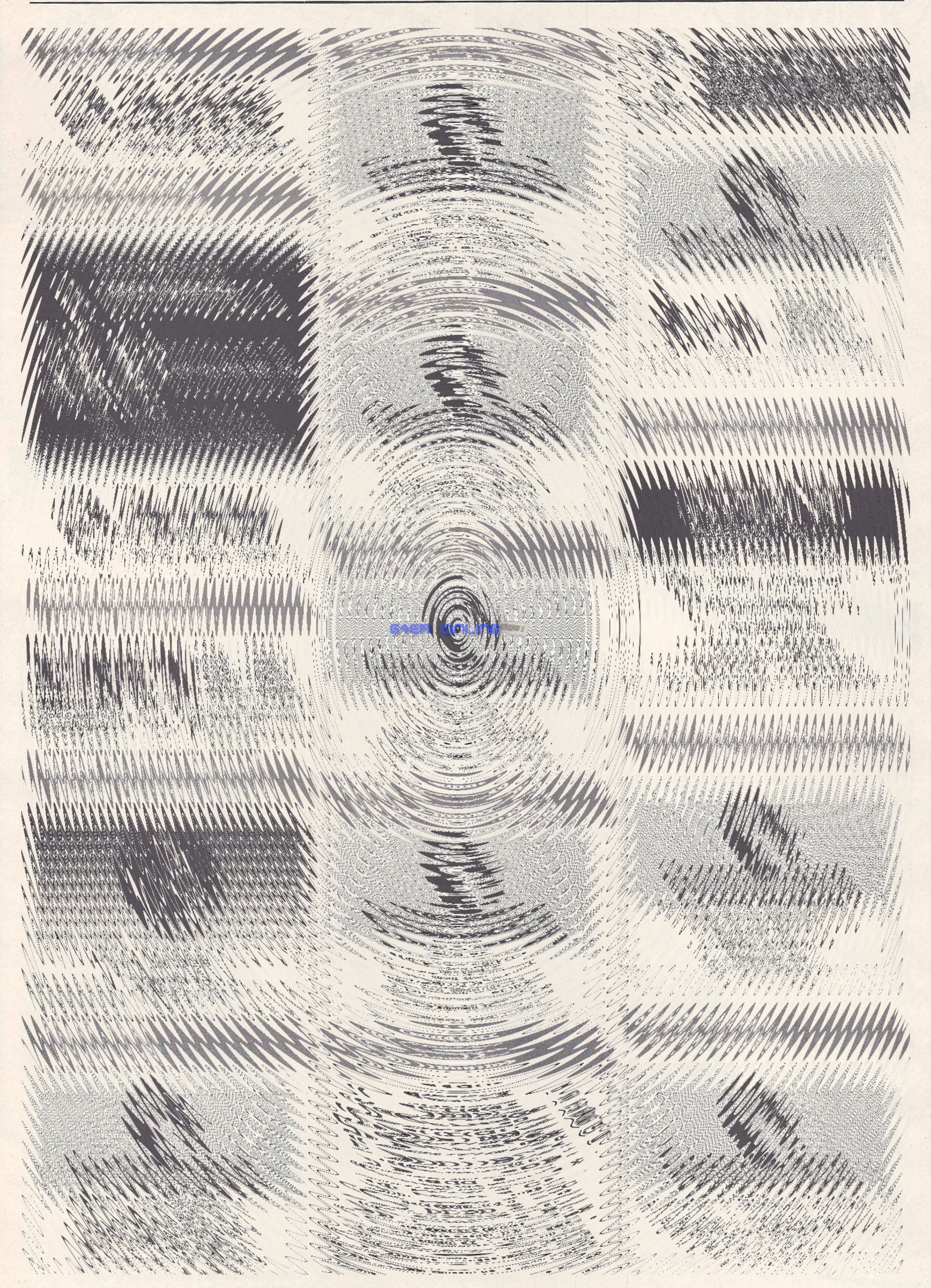
```
      N !":PRINT                                      <252>
5520 PRINT"{BLUE}WEITERE BUCHUNGEN SUCHEN (J/N)
      ?"                                              <114>
5530 : GET X$: IF X$<>"J" AND X$<>"N" GOTO 5530     <176>
5540 ::IF X$="J"AND S%<>0 GOTO 5180                  <134>
5550 :IF X$="N"THEN CLOSE 3:RETURN                   <138>
5560 PRINT CHR$(145)"{BLUE}GLEICHES KONTO{SPACE
     ,BROWN}"KN$(KN),"{BLUE,SPACE}(J/N) ?";          <167>
5570 :GET X$:IF X$<>"J"AND X$<>"N"GOTO 5570         <220>
5580 :IF X$="J"GOTO 5180                             <185>
5590 KN=0                                            <102>
5600 PRINT CHR$(147):GOSUB 3110:IF KN$="E"OR AK
     =0 THEN RETURN                                  <222>
5610 IF KN=ZW THEN 5180: REM PRUEFEN,                <005>
5620 ZW=KN:REM                   OB GLEICHE KN WIE
     VORHER                                          <164>
5630 CLOSE 3:GOSUB 6330:IF AA$="D"THEN A$=AA$:G
     OTO 5140                                        <015>
5640 GOTO 5180                                       <117>
5650 ::REM =*=* BUCHUNG AUF BILDSCHIRM               <194>
5660 ::REM ============================              <225>
5670 :                                               <118>
5680 PRINT D$(Y,0);"{BLUE}";                         <038>
5690 D=VAL(D$(Y,1)):SU=SU+D                          <245>
5700 PRINT TAB(29-LEN(D$(Y,1)));D$(Y,1);"{BROWN
     }";                                             <081>
5710 IF LEN(D$(Y,2))=5 GOTO 5730                     <178>
5720 PRINT TAB(31);MID$(D$(Y,2),3,2)"."LEFT$(D$
     (Y,2),2)"."RIGHT$(D$(Y,2),2):GOTO 5740         <000>
5730 PRINT TAB(31);MID$(D$(Y,2),2,2)". "LEFT$(D
     $(Y,2),1)"."RIGHT$(D$(Y,2),2)                   <117>
5740 ZZ=ZZ+1                                         <119>
5750 IF Y=Z-1 AND ZZ>=11 THEN ZZ=0:GOSUB 5780:G
     OTO 5770                                        <238>
5760 IF ZZ>14 AND Y<Z-1 THEN ZZ=0:GOSUB 5780:RE
     M =*= AUSGEBEN BIS 15 STUECK                    <126>
5770 RETURN                                          <046>
5780 PRINT TAB(20)"----------":PRINT TAB(10);"{
     LIG.BLUE}SUMME{BLUE,SPACE}";:GOSUB 4370         <006>
5790 PRINT TAB(26-(L-1))SU$;"{BROWN}"               <133>
5800 PRINT:PRINT"{LIG.BLUE}WEITER MIT{SPACE,RED
     ,RVSON,SPACE}TASTE{SPACE,RVOFF}";               <062>
5810 : WAIT 198,1:POKE 198,0                         <098>
5820 ::PRINT CHR$(147);:GOSUB 4990:PRINT"{RED}"
     KN$(KN)" "J$"{BROWN}":PRINT                     <163>
5830 RETURN                                          <106>
5840 : REM ** AUSGABE AUF DRUCKER **                 <059>
5850 : REM ============================              <101>
5860 D=VAL(D$(Y,1)):SU=SU+D                          <159>
5870 PRINT#4,CHR$(27)"D";                            <064>
5880 PRINT#4,CHR$(1)+CHR$(34-LEN(D$(Y,1)))+CHR$
     (39)+CHR$(0);                                   <001>
5890 PRINT#4,CHR$(9);D$(Y,0);CHR$(9);D$(Y,1);       <124>
5900 IF LEN(D$(Y,2))=5 GOTO 5930                     <115>
5910 PRINT#4,CHR$(9)MID$(D$(Y,2),3,2)"."LEFT$(D
     $(Y,2),2)"."RIGHT$(D$(Y,2),2)                   <113>
5920 GOTO 5940                                       <114>
5930 PRINT#4,CHR$(9)MID$(D$(Y,2),2,2)". "LEFT$(
     D$(Y,2),1)"."RIGHT$(D$(Y,2),2)                  <131>
5940 RETURN                                          <217>
5950 :                                               <143>
5960 PRINT#4,SPC(25)"----------":PRINT#4,SPC(10
     );"{LIG.BLUE}SUMME{BLUE,SPACE}";:GOSUB 437
     0                                               <132>
5970 PRINT#4,SPC(15-(L-1));SU$;"{BROWN}":CLOSE
     4:A$="":RETURN                                  <064>
5980 :                                               <173>
5990 REM ============================                <184>
6000 REM ==    DATEI   SORTIEREN    ==               <044>
6010 REM ============================                <204>
6020 :                                               <213>
6030 GOSUB 6300:IF KN$="E"OR AK=0 THEN RETURN        <011>
6040 IF ST=-128 OR R=21 THEN RETURN                  <037>
6050 N=Z:FU=5:PRINT CHR$(147):GOSUB 4990             <243>
6060 IF Z=0 THEN PRINT:PRINT TAB(10)"{RED}"D$(Z
     ,0):FOR I=0 TO 2000:NEXT:RETURN                 <105>
6070 FOR I=0 TO 2                                    <079>
6080 ::PRINT TAB(8)"{RVSON,RED}"I+1"{RVOFF,BROW
     N,2SPACE}"FB$(I)                                <047>
6090 NEXT I:PRINT:S%=0                               <019>
6100 PRINT TAB(4)"{BLUE}WONACH SOLL SORTIERT WE
     RDEN ?"                                         <097>
6110 PRINT TAB(4)"GEBEN SIE BITTE DIE NUMMER AN
      !":PRINT                                       <181>
6120 PRINT TAB(4)"ZURUECK ZUM MENUE -->{SPACE,R
     ED}RETURN{BLUE,SPACE}!"                         <079>
6130 GET X$:IF X$=CHR$(13)THEN RETURN                <202>
6140 IF X$<"1"OR X$>"3"GOTO 6130                     <233>
6150 X=VAL(X$)-1:PRINT                               <105>
6160 PRINT TAB(4)"ICH SORTIERE, BITTE WARTEN SI
     E !":S%=1:REM KENNZ. DATEI SORT.                <172>
6170 IF X=0 THEN GOSUB 600:GOTO 6190                 <084>
6180 GOSUB 420                                       <095>
6190 PRINT TAB(4)"{BLUE}DATEI{SPACE,RED}"KN$(KN
     )"{BLUE,SPACE}WURDE",TAB(4)"NACH{SPACE,RED
     }",FB$(X)"{BLUE,SPACE}SORTIERT !"               <125>
6200 PRINT:PRINT TAB(4)"BUCHUNGSPOSTEN AUSGEBEN
      (J/N) ?"                                       <217>
6210 :GET X$:IF X$<>"J"AND X$<>"N"GOTO 6210         <087>
6220 :IF X$="J"THEN GOSUB 5180                       <231>
6230 :FU=5:GOSUB 4820                                <191>
6240 RETURN                                          <006>
6250 :                                               <188>
6260 REM ============================                <199>
6270 REM ==    DATEI   LADEN        ==               <228>
6280 REM ============================                <219>
6290 :                                               <228>
6300 GOSUB 2900:IF KN$="E"OR AK=0 THEN RETURN        <028>
6310 IF KN=ZW THEN RETURN:REM       PRUEFEN,         <132>
6320 IF S%<>0 OR AV<>0 THEN GOSUB 4820:S%=0:REM
      OB GLEICHE KN WIE VORHER                       <111>
6330 ZW=KN:N$=LEFT$(KN$(KN),12)                      <123>
6340 PRINT CHR$(147):FU=1:GOSUB 4990                 <126>
6350 PRINT:PRINT:PRINT"{BLUE}DATEI{SPACE,RED}"K
     N$(KN) "{BLUE}WIRD GELADEN !"                   <092>
6360 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN        <195>
6370 OPEN 2,8,15                                     <193>
6380 OPEN 3,8,3,J$+N$+",S,R"                         <058>
6390 INPUT#2,R:IF R<>62 GOTO 6410                    <121>
6400 ::PRINT"DATEI NICHT GEFUNDEN !":CLOSE 2:CL
     OSE 3:FOR T=0 TO 3000:NEXT:RETURN               <169>
6410 Z=0:INPUT#3,D$(Z,0),D$(Z,1):IF ST=64 THEN
     CLOSE 3:CLOSE 2:RETURN                          <093>
6420 CLOSE 3                                         <000>
6430 OPEN 3,8,3,J$+N$+",S,R"                         <109>
6440 INPUT#3,D$(0,0)                                 <105>
6450 Z=1                                             <136>
6460 :                                               <143>
6470 ::INPUT#3,D$(Z,0),D$(Z,1),D$(Z,2)               <094>
6480 :                                               <163>
6490 IF ST<>64 THEN Z=Z+1:GOTO 6460                  <238>
6500 CLOSE 3:CLOSE 2:RETURN                          <036>
6510 :                                               <193>
6520 REM ============================                <204>
6530 REM ==    DATEI SICHERN        ==               <145>
6540 REM ============================                <224>
6550 :                                               <233>
6560 IF Z>0 GOTO 6590                                <220>
6570 POKE 214,23:POKE 211,8:SYS 58732:PRINT"KEI
     NE DATEN IM RECHNER"                            <184>
6580 :FOR X=0 TO 2000:NEXT:RETURN                    <172>
6590 PRINT CHR$(147);:FU=2:GOSUB 4990                <181>
6600 PRINT:PRINT"{BLUE}DISKETTE EINGELEGT ?":PR
     INT                                             <179>
6610 PRINT"{BROWN}DRUECKEN SIE ANSCHLIESSEND{SP
     ACE,RED,RVSON,SPACE}S{SPACE,RVOFF,BLUE}":P
     RINT:FOR I=0 TO 800:NEXT                        <054>
6620 PRINT:PRINT"WOLLEN SIE DIE GEAENDERTE DATE
     I ",KN$(ZW)" NICHT SICHERN{2SPACE}?"            <084>
6630 PRINT"DANN MENUE{3SPACE}--->{3SPACE,RED}RE
     TURN{BLUE}"                                     <087>
6640 GET X$:IF X$<>"S"AND X$<>CHR$(13)GOTO 6640     <198>
6650 : IF X$=CHR$(13)THEN RETURN                     <183>
6660 GOSUB 330:IF ST=-128 OR R=21 THEN RETURN        <241>
6670 PRINT:PRINT"BUCHUNGSDATEI{SPACE,RED}"KN$(Z
     W)"{BLUE,SPACE}WIRD","GESICHERT !"              <065>
6680 N$=LEFT$(KN$(ZW),12)                            <189>
6690 CLOSE 3:OPEN 3,8,15,"S:"+J$+N$                  <066>
6700 CLOSE 3:OPEN 3,8,3,J$+N$+",S,W"                 <142>
6710 : PRINT#3,D$(0,0)                               <198>
6720 : FOR Y=1 TO Z                                  <079>
6730 :    FOR I=0 TO 2                               <032>
6740 :       PRINT#3,D$(Y,I)                         <038>
6750 :    NEXT I                                     <125>
6760 : NEXT Y                                        <151>
6770 CLOSE 3                                         <095>
6780 PRINT:AV=0:S%=0                                 <118>
6790 PRINT"BUCHUNGSDATEI{SPACE,RED}"KN$(ZW)"{BL
     UE,SPACE}IST{2SPACE}GESICHERT !":PRINT          <003>
6800 PRINT:PRINT"WEITER MIT{SPACE,RED}TASTE{BLU
     E,SPACE}!"                                      <241>
6810 WAIT 198,1:POKE 198,0                           <020>
```

Listing 1. Das Haushaltsbuch (Fortsetzung)

```
6820 RETURN                                    <076>
6830 :                                         <002>
6840 REM ============================          <013>
6850 REM == BUCHUNGEN AENDERN       ==         <251>
6860 REM ============================          <033>
6870 :                                         <042>
6880 GOSUB 6300:IF KN$="E"OR AK=0 THEN RETURN  <096>
6890 IF ST=-128 OR R=21 THEN RETURN            <122>
6900 PRINT CHR$(147):FU=4:GOSUB 4990           <179>
6910 IF Z=0 THEN PRINT:PRINT TAB(9)"{RED}"D$(Z,
     0):FOR I=0 TO 2000:NEXT:RETURN            <150>
6920 ZZ=0:ZS=10:AV=0:PRINT                     <057>
6930 PRINT"{BLUE}WENN SIE GEZIELT BUCHUNGPOSITI
     ONEN ODER MONATE AENDERN WOLLEN,";         <141>
6940 PRINT" GEBEN SIE BITTE","{SPACE,RED,RVSON}
     E{BLUE,RVOFF,2SPACE}EIN."                 <108>
6950 PRINT:PRINT"{RED}RETURN{LIG.BLUE,SPACE}LIS
     TET DIE DATEI VOLLSTAENDIG{11SPACE}AUF !" <030>
6960 GET E$:IF E$<>"E"AND E$<>CHR$(13)GOTO 6960 <198>
6970 IF E$="E"THEN GOSUB 5190:GOTO 7060        <023>
6980 :                                         <153>
6990 REM =*= ALLE POS. AUFLISTEN =*=           <041>
7000 REM ===========================           <113>
7010 PRINT CHR$(147);:GOSUB 4990               <162>
7020 GOSUB 7780                                <234>
7030 FOR Y=1 TO Z                              <076>
7040 :   GOSUB 7150                            <047>
7050 NEXT Y                                    <128>
7060 PRINT:PRINT"{BLUE}DATEIENDE ERREICHT !"   <120>
7070 IF E$=CHR$(13)THEN 7280                   <083>
7080 IF ZZ<>0 AND E$="E"THEN E$="":GOTO 7280   <131>
7090 :::FOR I=1 TO 2000:NEXT                   <074>
7100 RETURN                                    <101>
7110 :                                         <027>
7120 REM =*= AUSGABE AUF BILDSCHIRM ==         <077>
7130 REM ===-----------------------===         <253>
7140 GOSUB 7780:REM BEI EINZEL-SELEKT.         <229>
7150 IF Y<11 AND E$<>"E"THEN PRINT SPC(3-LEN(ST
     R$(Y)))"{RED}"Y"{BROWN}"D$(Y,0)"{BLUE}";:G
     OTO 7170                                  <166>
7160 PRINT SPC(4-LEN(STR$(Y)))"{RED}"Y"{BROWN}"
     D$(Y,0)"{BLUE}";                          <122>
7170 PRINT TAB(30-LEN(D$(Y,1)));D$(Y,1)"{ORANGE
     }";                                       <190>
7180 IF LEN(D$(Y,2))=5 GOTO 7210               <113>
7190 PRINT TAB(31);MID$(D$(Y,2),3,2)"."LEFT$(D$
     (Y,2),2)"."RIGHT$(D$(Y,2),2)              <049>
7200 GOTO 7220                                 <112>
7210 PRINT TAB(31);MID$(D$(Y,2),2,2)". "LEFT$(D
     $(Y,2),1)"."RIGHT$(D$(Y,2),2)             <067>
7220 ZZ=ZZ+1                                   <069>
7230 :IF ZZ=ZS AND ZZ<Z THEN ZS=ZS+10:GOSUB 728
     0                                         <248>
7240 RETURN                                    <242>
7250 :                                         <168>
7260 REM =*= AENDERUNGSABFRAGE =*=             <035>
7270 :                                         <188>
7280 PRINT:PRINT"{RED}AENDERUNGEN (J/N) ?"     <219>
7290 : GET X$:IF X$<>"J"AND X$<>"N" THEN 7290  <186>
7300 : IF X$="N"AND Y<Z THEN PRINT CHR$(147):GO
     SUB 4990:GOSUB 7780:RETURN                <048>
7310 : IF X$="N"AND Y>=Z AND AV=1 THEN GOSUB 65
     90:RETURN:REM AEND.SICHERN                <008>
7320 : IF X$="N"AND Y>=Z AND AV=0 THEN RETURN:R
     EM KEINE AENDERUNG-->MENUE                <059>
7330 :PRINT CHR$(145)"{39SPACE}":REM 39 X LEERT
     .                                         <190>
7340 :                                         <002>
7350 REM =*= AENDERUNG          =*=            <066>
7360 REM =========================             <096>
7370 C=17:S=0:GOSUB 830                        <020>
7380 PRINT"{LIG.BLUE}GEBEN SIE BITTE DIE ZU AEN
     DERNDE POSI-{2SPACE}TION EIN (ZEILE, SPALT
     E) !"                                     <134>
7390 :INPUT A,B:B=B-1:IF B<>0 AND B<>2 GOTO 737
     0                                         <048>
7400 : IF A<1 OR A>Y GOTO 7370                 <067>
7410 : IF Y-1=Z AND A>Y-1 OR B>2 OR B<0 THEN C=
     17:S=0:GOSUB 830:GOTO 7380                <254>
7420 : IF B<>2 GOTO 7470                       <016>
7430 : IF LEN(D$(A,2))=5 GOTO 7460             <149>
7440 : PRINT TAB(2);MID$(D$(A,2),3,2)"."LEFT$(D
     $(A,2),2)"."RIGHT$(D$(A,2),2)             <236>
7450 : GOTO 7480                               <205>
7460 : PRINT TAB(2);MID$(D$(A,2),2,2)". "LEFT$(
     D$(A,2),1)"."RIGHT$(D$(A,2),2)            <254>
7470 IF B<>2 THEN PRINT"{BROWN,2SPACE}";D$(A,B) <106>
7480 PRINT:PRINT"BITTE AENDERN SIE !{20SPACE}":
     AV=1                                      <048>
7490 C=20:S=0:GOSUB 830                        <135>
7500 :                                         <163>
7510 INPUT D$(A,B):PRINT"{2SPACE,RED}"D$(A,B):I
     F LEN(D$(A,B))>20 THEN D$(A,B)=LEFT$(D$(A,
     B),20)                                    <031>
7520 IF B<>2 GOTO 7590:REM DATUM PRUEFEN       <151>
7530 IF MID$(D$(A,2),3,1)="."AND MID$(D$(A,2),6
     ,1)="."GOTO 7570                          <198>
7540 IF MID$(D$(A,2),2,1)="."AND MID$(D$(A,2),4
     ,1)="."THEN GOSUB 7710:GOTO 7590          <012>
7550 IF MID$(D$(A,2),2,1)="."AND MID$(D$(A,2),5
     ,1)="."THEN GOSUB 7720:GOTO 7590          <024>
7560 IF MID$(D$(A,2),3,1)="."AND MID$(D$(A,2),5
     ,1)="."THEN GOSUB 7730:GOTO 7590          <036>
7570 D$(A,2)=MID$(D$(A,2),4,2)+LEFT$(D$(A,2),2)
     +RIGHT$(D$(A,2),2)                        <221>
7580 :                                         <243>
7590 PRINT" WEITERE AENDERUNGEN (J/N) ?"       <056>
7600 : GET X$:IF X$<>"J"AND X$<>"N"THEN 7600   <236>
7610 ::IF X$="N"AND Y<Z THEN PRINT CHR$(147):GO
     SUB 4990:GOSUB 7780:RETURN                <161>
7620 :IF X$="N"AND Y>=Z AND AV=1 THEN GOSUB 659
     0:RETURN:REM AEND.SICHERN                 <063>
7630 C=19:S=1:GOSUB 830                        <028>
7640 PRINT"{BLUE,8SPACE}"                      <241>
7650 PRINT"{39SPACE}":REM 39 X LEERTASTE       <018>
7660 PRINT"{39SPACE}"                          <230>
7670 C=17:S=0:GOSUB 830                        <065>
7680 GOTO 7380                                 <121>
7690 :                                         <098>
7700 REM =*= DATUM AUFBEREITEN MMTTJJ *        <010>
7710 D$(A,2)=MID$(D$(A,2),3,1)+"0"+LEFT$(D$(A,2
     ),1)+RIGHT$(D$(A,2),2):RETURN             <077>
7720 D$(A,2)=MID$(D$(A,2),3,2)+"0"+LEFT$(D$(A,2
     ),1)+RIGHT$(D$(A,2),2):RETURN             <088>
7730 D$(A,2)=MID$(D$(A,2),4,1)+LEFT$(D$(A,2),2)
     +RIGHT$(D$(A,2),2):RETURN                 <069>
7740 :                                         <148>
7750 REM =*= SPALTEN-BESCHRIFTUNG =*=          <003>
7760 REM ===========================           <108>
7770 :                                         <178>
7780 PRINT TAB(5)"{RED}1";TAB(26)"2";TAB(33)"3"
     :PRINT:RETURN                             <017>
7790 :                                         <198>
7800 REM ==============================        <075>
7810 REM ** DATAS FUER INPUT-MODIFIKAT.        <165>
7820 REM ==============================        <095>
7830 FOR I=828 TO 922                          <019>
7840 READ X:POKE I,X:S=S+X:NEXT I              <212>
7850 DATA 169,71,160,3,141,8,3,140,9,3,96,32   <215>
7860 DATA 115,0,201,133,240,6,32,121,0,76,231,1
     67                                        <192>
7870 DATA 32,115,0,201,172,240,6,32,191,171,76,
     174                                       <010>
7880 DATA 167,32,155,183,32,30,225,32,253,174,3
     2,158                                     <134>
7890 DATA 183,138,72,32,253,174,32,139,176,133,
     73,132                                    <202>
7900 DATA 74,32,163,182,104,32,117,180,160,2,18
     5,97                                      <105>
7910 DATA 0,145,73,136,16,248,200,32,18,225,145
     ,98                                       <066>
7920 DATA 200,196,97,208,246,32,204,255,76,174,
     167                                       <095>
7930 IF S<>11096 THEN PRINT"FEHLER IN DATAS!!":
     END                                       <149>
7940 SYS 828:RETURN                            <043>
7945 REM ==============================        <221>
7950 REM * SWAP - FUNKTION SIMULIEREN *        <227>
7960 REM ==============================        <236>
7970 REM SYNTAX:SYS715(STRING1,STRING2)        <027>
7980 REM ==============================        <000>
7990 DATA 32,250,174,32,158,173,32,143,173,165,
     100,133,247,165,101,133                   <096>
8000 DATA 248,32,253,174,32,158,173,32,143,173,
     160,0,177,247,133,249                     <030>
8010 DATA 177,100,145,247,165,249,145,100,200,1
     92,3,208,239,32,247,174,96                <031>
8030 SA=715:SE=SA+48:FOR I=SA TO SE:READ X:POKE
      I,X:CS=CS+X:NEXT                         <161>
8040 IF CS<>7314 THEN PRINT"FEHLER IN DATA'S":E
     ND                                        <250>
8050 RETURN                                    <031>
```

Listing 1. Das Haushaltsbuch (Schluß)


64ER ONLINE
64er-online.de
64er-online.net





HYPRA-ASS — Ein Assembler der Spitzenklasse

Machen Sie es wie die Profis. Schreiben Sie Programme in Maschinensprache. Dieser leistungsstarke Makro-Assembler macht es möglich.

HYPRA-ASS ist ein rein in Maschinenesprache geschriebener Drei-Pass-Makroassembler mit integriertem Editor für den C 64 mit Floppy 1541. Er wird mit LOAD "HYPRA-ASS",8 geladen und durch RUN gestartet. Nach dem Start meldet sich HYPRA-ASS mit »break in 0« und »ready«. Alle Basic-Befehle sind nach dem Start noch zu verwenden, bis auf die Befehle LET und FOR, die Variable anlegen. Der Befehl RUN dient jetzt zum Starten der Assemblierung.

Der Quelltext

Der Quelltext wird vom HYPRA-ASS-Editor in Basic-Programmzeilen abgelegt. Soweit wie möglich werden unnötige Blanks dabei eleminiert. Für die einzelnen Quelltextzeilen gelten die folgenden Vereinbarungen:

1. Bei der Eingabe einer Zeile wird hinter der Zeilennummer ein Minuszeichen eingegeben.
2. Jede Zeile enthält höchstens einen Assemblerbefehl.
3. Vor einem Assemblerbefehl darf in derselben Zeile höchstens ein Label stehen.
4. Label beginnen direkt hinter dem Minuszeichen.
5. Vor jedem Assemblerbefehl steht mindestens ein Blank.
6. Label und Assemblerbefehl werden durch mindestens ein Blank voneinander getrennt.
7. Ein Label darf nicht allein in einer Zeile stehen.
8. Kommentar wird durch ein Semikolon vom Rest der Zeile getrennt.
9. Reine Kommentarzeilen müssen als erstes Zeichen hinter dem Minuszeichen ein Semikolon haben.
10. Pseudo-Ops (.ba, .eq...) können direkt hinter dem Minuszeichen beginnen.

Beispiele:

```
100  -.ba $C000
110  -initialisierung
120  -; reine Kommentarzeile
130  -            lda $14; Kommentar hinter einem Befehl
140  -marke       ldx $15; mit Label davor
```

Zur bequemeren Eingabe und Bearbeitung des Quelltextes stellt HYPRA-ASS im Editor insgesamt 25 Befehle zur Verfügung (Bild 1).

Rechnungen im Quelltext

HYPRA-ASS erlaubt die vier Grundrechenarten plus Potenzierung, die logischen Operationen NOT, AND und OR, die Vergleiche »gleich«, »kleiner« und »größer«, sowie den Einsatz der Funktionen <(...) und >(...), die das Low- beziehungsweise Highbyte eines Argumentes liefern. Die logischen Operatoren und die Vergleiche werden wie folgt abgekürzt:

!n! = not !a! = and !o! = or != ! = gleich
!<! = kleiner als !>! = größer als

Das Ergebnis eines Vergleiches ist —1, falls wahr, 0, falls nicht wahr: (1!=!2)=0 (1!=!1)=—1

Auch die NOT-Verknüpfung arbeitet wie in Basic: !n!1 = —2. Das Argument in den Low-/Highbyte-Funktionen muß im Bereich 0 ≤ Argument ≤65535 liegen.

Außer Dezimalzahlen sind Hex-Zahlen erlaubt, die durch ein vorangestelltes Dollarzeichen kenntlich gemacht werden: $C000 = 49152 $10 = 16 $a = 10...

Die Hexzahlen können auch in den Basic-Befehlen verwendet werden.

HYPRA-ASS-Variable (Label)

Der Wert einer HYPRA-ASS-Variablen kann zwischen 0 und $FFFF liegen. Variablennamen können beliebig lang sein, wobei das erste Zeichen des Variablennamens ein Buchstabe sein muß. Weitere Zeichen können Buchstaben, Ziffern oder das Hochkomma sein. Alle Zeichen des Namens sind signifikant.

Im Zusammenhang mit der Verwendung von Makros muß zwischen globalen und lokalen Variablen unterschieden werden. Jede Variable erhält beim Anlegen eine sogenannte Ord-

/a 100,10	Automatische Zeilennumerierung. Hier mit der Startnummer 100 und der Schrittweite 10. Die automatische Zeilennumerierung wird ausgeschaltet, indem man direkt hinter dem ausgegebenen Minuszeichen RETURN eingibt.
/o	Re-New eines Quelltextes, der mit NEW gelöscht wurde, falls der Text nicht anderweitig zerstört wurde.
/d ; /d 100 ; /d —100 ; /d 100— ; /d 100 bis 200	Löschen von Zeilen und Zeilenbereichen. Auch für das Löschen einzelner Zeilen sollte man den /d-Befehl verwenden, da man das Minuszeichen hinter der Zeilennummer doch immer wieder vergißt.
/e ; /e 100 ; /e —100 ; /e 100— ; /e 100 bis 200	Formatiertes Listen von Zeilen und Zeilenbereichen. Label, Assembler-Befehle werden gemäß den Tabulatoren übersichtlich untereinander geschrieben.
/t0,13 ; /t 1,24 ; /t2,0 ; /t3,10	setzt die Tabulatoren T0, T1, T2, T3 T0 = Tabulator für Assemblerbefehle T1 = Tabulator für den Kommentar bei der formatierten Ausgabe T2 = Tabulator für die Anzahl der Blanks, die am Anfang einer Ausgabezeile ausgegeben werden T3= Tabulator für die Symboltabelle
/x	Verlassen des Assemblers. Beim Verlassen des Programms wird ein Reset durchgeführt.
/p 1,100,200	Setzen eines Arbeitsbereichs (Page). Hier Bereich 1 von Zeile 100 bis 200, beide einschließlich. Bis zu 30 solcher Arbeitsbereiche sind erlaubt. Die Parameter der Arbeitsbereiche werden im Kassettenpuffer abgelegt.
/ziffer(n)	Formatiertes Listen der Page.
/n 1,100,10	Neu Durchnumerieren einer Page mit Startnummer und Schrittweite.
/f 1,"string"	Suchen einer Zeichenkette in einer Page. Dabei sind im String Fragezeichen als Joker erlaubt. Das Fragezeichen ersetzt ein beliebiges Zeichen. Zu beachten ist jedoch, daß im Quelltext unnötige Blanks entfernt wurden, wie ein Vergleich mit den Befehlen /e und LIST zeigt.
/r 1,"string1", "string2"	Ersetzen von Zeichenketten. String 2 darf nicht leer sein. Überall in der Page wird die Zeichenkette aus String 2 durch die aus String1 ersetzt. Auch beim Ersetzen ist in String 2 das Fragezeichen als Joker erlaubt. Da String 1 leer sein darf, können mit diesem Befehl auch Zeichenketten gelöscht werden.
/u 9000	Setzen des Quelltextstartes (Programmstartes). Normalerweise ist als Startwert die Adresse 7000 ein-

nungszahl, die tatsächlich angibt, im wievielten Makroaufruf das Anlegen stattfand. Befinde ich mich in gar keinem Makro, ist die Ordnungszahl entsprechend Null.

Variable mit unterschiedlicher Ordnungszahl sind trotz gleichen Namens nicht gleich. Man kann also davon sprechen, daß Variable gleicher Ordnungszahl lokal sind.

Die Konstruktion mittels Ordnungszahlen dient dazu, Fehler durch doppelte Benutzung von Labeln bei mehrmaligem Aufruf von Makros zu verhindern, indem Makros bei jedem Aufruf sozusagen einen komplett neuen Satz von Labeln erhalten.

Andererseits sind aus einem Makro »herausgesehen« alle Variablen mit anderer Ordnungszahl als im Makro selbst »unsichtbar«. Um aber bequem Makros in Makros aufrufen und bequem Label verwenden zu können, die in mehreren Makros benutzt werden sollen (etwa Betriebssystemroutinen), gibt es die globalen Variablen.

Globale Variable sind, wie der Name schon verrät, im Gegensatz zu den lokalen Variablen unabhängig von der Ordnungszahl überall definiert.

Alle Makronamen sind per Definition global.

	gestellt. Durch Hochlegen des Startes kann man zum Beispiel einen Monitor in dem nun freien Bereich unterbringen.
/b	Anzeige der aktuellen Speicherkonfiguration. Es wird angezeigt: a) der normale Quelltextstart 7000 als Merkhilfe b) der aktuelle Quelltextstart c) das Quelltextende d) die Anzahl der noch verbleibenden Bytes für den Quelltext
/l"name" ; /s"name" ; /v"name" ; /m"name"	Kurzform der Befehle LOAD, SAVE, VERIFY und MERGE
/g 8	Die zugehörige Gerätenummer kann mit diesem Befehl eingestellt werden. Voreingestellt ist das Gerät 8.

Zur Unterstützung des Umgangs mit dem Floppy-Laufwerk 1541 sind drei Befehle implementiert:

/i	— Lesen des Inhaltsverzeichnisses von Floppy ohne Verlust des geschriebenen Quelltextes
/k	— Lesen des Fehlerkanals
/e	— Übermittlung von Befehlen an die Floppy

Diese drei Befehle entsprechen denen des DOS 5.1.

Auch zur Farbgebung des Bildschirms sind zwei Befehle vorhanden, die die Hintergrund- und die Rahmenfarbe setzen.

/ch 0	— Setzen der Hintergrundfarbe
/cr 0	— Setzen der Rahmenfarbe

Nach erfolgter Assemblierung kann nun die erzeugte Symboltabelle mit zwei Befehlen ausgegeben werden:

/!	— Ausgabe in unsortierter Form
/!!	— Ausgabe sortiert

Es werden nur Label ausgegeben, die entweder global oder von der Ordnung Null sind.

Beide Dumps können mit der CTRL-Taste verlangsamt und mit der STOP-Taste angehalten werden.

Mit OPEN..., CMD... können die Dumps an andere Geräte gesendet werden.

Als Ergänzung zum Basic-Befehl PRINT, der aufgrund der Torkenbildung nicht alle Labelnamen verarbeiten kann, kann der Befehl ← verwendet werden.

Basic-Funktionen wie PEEK sind nur über den PRINT-Befehl erreichbar. Die Funktionen <(...) und >(...) sind außerhalb des Quelltextes nur durch ← zu verwenden. Mit dem ←-Befehl kann genau wie im Quelltext gerechnet werden.

Bild 1. Die Editorbefehle

Alle Variablen sind bei HYPRA-ASS redefinierbar gehalten, das heißt alle Variablen können durch eine Wertzuweisung jederzeit verändert werden.

Eine dopelte Benutzung von Labeln vor Assemblerbefehlen wird jedoch durch einen »Label twice«-Error (Bild 2) geahndet, da dies zu einem falschen Ergebnis der Assemblierung führen würde.

Die Makros von HYPRA-ASS

Makros sind meist kürzere Befehlsfolgen, die im Quelltext häufiger vorkommen, und deshalb unter einem Makro zusammengefaßt werden. Zu jedem Makro gehört ein Makroname, mit dem es aufgerufen werden kann. An jedes HYPRA-ASS-Makro können beliebig viele Parameter übergeben werden, deren aktueller Wert dann bei der Assemblierung im Makro eingesetzt wird. Makros können bei HYPRA-ASS an beliebiger Stelle im Quelltext definiert werden. Alle Makronamen sind global, alle Parameter und makrointernen Label sind lokal. Das heißt verschiedene Makros können durchaus Label beziehungsweise Parameter gleichen Namens verwenden.

Ein Beispiel für ein einfaches Makro:

Es wird immer wieder die Befehlsfolge benötigt, Akkumulator und X-Register mit dem Inhalt zweier aufeinanderfolgender Speicherzellen zu laden. Ein Makro dazu könnte folgendermaßen aussehen:

```
100  -.ma ldax (adresse)
110  -          lda adresse
120  -          ldx adresse+1
130  -.rt
```

Der .ma-Pseudobefehl wird gefolgt von einem Variablennamen, dem Makronamen, und einer Parameterliste in runden Klammern, falls Parameter vorhanden sind. Hier ist es ein Parameter, die Adresse der Speicherzelle, die in den Akku soll. Sind mehrere Parameter vorhanden, werden sie durch Kommata getrennt. In die Parameter setzt der Assembler bei jedem Aufruf den aktuellen Wert, der im Aufruf steht. Rufe ich also ldax (2) auf, so entsteht bei der Assemblierung des Makro die Folge lda 2, ldx3, entsprechend führt der Aufruf mit ldax (label) zu lda label, ldx label+1.

Die Parameterliste darf in der Definitionszeile eines Makro nur aus einer Folge von Variablennamen bestehen, während im Aufruf als aktuelle Parameter beliebige Ausdrücke erlaubt sind. Hinter der Definitionszeile mit dem .ma-Pseudo folgt dann der eigentliche Makroinhalt, das heißt das, was bei einem Aufruf des Makro assembliert werden soll.

Natürlich sind hier nicht nur einfache Befehle wie im Beispiel gestattet. Genausogut können im Makro Verzweigungen und Sprünge ausgeführt werden, es kann bedingt assembliert werden, und weitere Makros können aufgerufen werden. Für die Schachtelung von Makros besteht keine Grenze außer der Fassungskapazität des Prozessorstacks.

Als Beispiel — wird ein Makro mit zehn internen Labeln 100 mal aufgerufen, ergibt sich schon für die dadurch erzeugten lokalen Label ein Platzbedarf von genau 7 000 Byte.

Sollte irgendwann der Fall eintreten, daß Label und Quelltext zusammen nicht mehr ins RAM passen, erhalten Sie den »too many labels«-Error (Bild 2). Dies ist allerdings mehr ein theoretischer Fall, denn auch bei der Assemblierung von HYPRA-ASS selbst wurden trotz extensiver Benutzung von Labels nicht einmal 500 gebraucht. Sie können aber davon ausgehen, daß Ihnen immer mindestens Platz für 1 170 Label zur Verfügung steht — in den allermeisten Fällen sogar erheblich mehr.

Selbstaufrufe von Makros sind auch nicht verboten. Inwieweit eine solche Konstruktion überhaupt sinnvoll sein kann, bleibt jedem selbst zu prüfen.

Zurück zur Makrodefiniton: Jede Makrodefinition muß unbedingt mit dem Pseudo .rt (return) abgeschlossen sein. Trifft der Assembler bei der Abarbeitung eines Makro auf .rt, so heißt das für ihn, die Assemblierung hinter dem Aufruf fortzusetzen. Vor der .ma und .rt-Anweisung dürfen in derselben Zeile keine Label stehen. Die Makrodefinition selbst wird in Pass 1 und Pass 2 überlesen. Es zählen also nur die Makroaufrufe bei der Assemblierung.

Der Aufruf eines Makro erfolgt durch den Pseudobefehl ..., gefolgt vom Makronamen und der aktuellen Parameterliste in runden Klammern.

Wertzuweisung an Label

Zwei Pseudobefehle stehen zur Verfügung, um Label einen Wert zuzuweisen:

.eq — weist einem Label einen Wert zu, ohne die Ordnungszahl des Labels dabei zu verändern.

.gl — erklärt gleichzeitig das Label als global.

Beide Pseudos werden der eigentlichen Wertzuweisung vorangestellt, so wie LET in Basic:

```
100   -.eq marke = $FFC0
110   -.gl label  = $200
```

Zusätzlich zu den Fehlermedlungen, die von Interpreterroutinen wie »illegal quantity« oder »syntax« stammen, gibt HYPRA-ASS folgende Meldungen aus:

1. **can't number term** — ein Ausdruck kann von HYPRA-ASS nicht berechnet werden. Möglicher Grund kann die falsche Abkürzung eines Operators sein.
2. **end of line expected** — bei der Abarbeitung einer Zeile wurde statt des Zeilenendes etwas anderes gefunden.
3. **no mnemonic** — ein Mnemonic kann nicht identifiziert werden.
4. **unknown pseudo** — ein Pseudo-Op wurde falsch abgekürzt.
5. **illegal register** — ein Assemblerbefehl existiert in der gewählten Adressierungsart nicht mit dem gewählten Register.
6. **wrong adress** — ein Assemblerbefehl existiert nicht in der gewählten Adressierungsart.
7. **illegal label** — das erste Zeichen eines Labels war kein Buchstabe.
8. **unknown label** — in Pass 2 wurde ein unbekannter Labelname entdeckt
9. **branch too far** — eine Verzweigung führt über eine zu große Distanz.
10. **label declared twice** — ein Labelname wurde zweimal benutzt.
11. **too many labels** — Label und Quelltext passen zusammen nicht mehr in den Speicher.
12. **no macro to close** — die Anzahl der .ma-Anweisungen stimmt nicht mit der Anzahl der .rt-Anweisungen überein.
13. **parameter** — im Makroaufruf stimmt die Parameterliste nicht mit der Parameterliste der Definition überein.
14. **return** — es liegt keine Rückkehradresse auf dem Stack, als eine .rt-Anweisung ausgeführt werden sollte.

Hinzuweisen ist noch auf eine einfache Möglichkeit, den »label twice-error« zu vermeiden:
Legt man eine Makrodefinition um einen beliebigen Block des Quelltextes, so sind alle Label in dem Block automatisch lokal. Auf diese Weise kann schon vorhandener Quelltext in neuen eingefügt werden, ohne daß man sich um doppelt verwendete Labelnamen kümmern muß.

Bild 2. Fehlermeldungen von HYPRA-ASS

Bei der Wertzuweisung an Label ist immer der Bereich einzuhalten in dem ein Labelwert liegen darf (0 bis $FFFF).

Einfügen von Tabellen und Text

Drei Pseudo-Ops erleichtern das Einfügen von Tabellen und Text in den Quelltext. Dies sind:

.by — erlaubt das Einfügen von Bytewerten (Werten zwischen 0 und $FF). Einzelne Bytewerte werden durch Kommata voneinander getrennt. Auch Strings der Länge 1 sind als Bytewerte erlaubt. Beispiel:
100 -.by 0, "a", 123, "x", $fa

.wo — erlaubt das Einfügen von Adressen (Werten zwischen 0 und $FFFF). Mehrere Adressen werden durch Kommata voneinander getrennt. Die Adressen werden in der Folge Low/Highbyte in den Objektcode aufgenommen. Beispiel:
100 -.wo marke-1, label*2-1

.tx — erlaubt das Einfügen von Text in den Quelltext. Die einzelnen Zeichen des Textes werden als ASCII-Code im Objektcode aufgenommen. Beispiel:
100 -.tx "beispieltext"

Überall im Quelltext, wo Bytewerte erwartet werden, etwa bei der unmittelbaren Adressierung, können Strings der Länge 1 verwendet werden. Ein Befehl lda#"a" ist also erlaubt.

Die bedingte Assemblierung

Zur Unterstützung der bedingten Assemblierung bietet HYPRA-ASS ein IF/ELSE/ENDIF-Konstrukt und ein IF/THEN-Konstrukt. Außerdem steht ein unbedingter Sprungbefehl zur Verfügung.

.on — entspricht dem IF/THEN von Basic. Hinter .on folgt ein Ausdruck, ein Komma und ein zweiter Ausdruck. Ist der erste Ausdruck wahr, wird zu der Zeilennummer gesprungen, die der zweite Ausdruck angibt. Beispiel:
100 -.on switch !=! 7, 400
Es wird die Assemblierung in Zeile 400 fortgesetzt, wenn switch gleich 7 ist.

.go — ergibt einen unbedingten Sprung zu der Zeile, die der Ausdruck hinter .go angibt. Beispiel:
100 -.go 1000

.if — wird gefolgt von einem Ausdruck. Ist der Ausdruck wahr, wird die Assemblierung hinter der .if-Zeile fortgesetzt, bis

.el — gefunden wird. Daraufhin wird

.ei — gesucht und dahinter die Assemblierung fortgesetzt.

Entsprechend erfolgt die Assemblierung von .el bis .ei, falls der Ausdruck hinter .if falsch ist. .el kann auch fehlen, es wird dann direkt hinter .ei fortgefahren.

Auf eine Schachtelung von IF-Konstrukten wurde wegen des Zwecks der bedingten Assemblierung verzichtet. Beispiel:

```
100   -.if switch !=! 6
110   -              lda#0
120   -.el
130   -              lda#2
140   -.ei
```

Wenn switch gleich 6 ist, erhält man lda#0, sonst wird lda#2 erzeugt. Vor den Pseudos .if, .el und .ei dürfen keine Label in derselben Zeile stehen.

Verkettung von Quelltexten

Mit dem Pseudo .ap (append) kann ein weiterer Quelltext am Ende des Pass 2 automatisch nachgeladen werden, wobei der Programmzähler aus der vorangegangenen Assemblierung erhalten bleibt.

Hinter .ap muß der Name des nachzuladenen Files in Anführungszeichen stehen.

Eine Besonderheit von HYPRA-ASS bildet im Zusammenhang mit verketteten Quelltexten der Pseudo-Opcode .co (common).

Dieser Befehl bewirkt zunächst, daß alle Variablen/Label, die hinter der .co-Anweisung in einer Liste stehen, an den nachgeladenen Teil übergeben werden.

Zweitens bleiben alle Quelltextzeilen bis zur common-Zeile beim Nachladen erhalten. Steht also etwa ein Makro vor der common-Zeile, wird auch das Makro übergeben. Zu beachten ist dabei:

a) Es sollten keine Makroaufrufe im common-Bereich stehen, es sei denn innerhalb eines Makro.

b) Die .ba-Anweisung, die die Startadresse des Objektcodes bestimmt, sollte außerhalb des common-Bereiches liegen, damit nach dem Nachladen nicht wieder mit der gleichen Startadresse assembliert wird.

c) Wertzuweisungen an Label sollten ebenfalls außerhalb des common-Bereiches liegen, um Platz für den nachgeladenen Quelltext zu gewinnen.

Direktes Senden des Objektcodes zur Floppy

Der Pseudobefehl .ob (object), gefolgt vom Filenamen ,p,w in Anführungszeichen, sendet den erzeugten Objektcode direkt zur Floppy.

Geschlossen wird das so erzeugte Objektfile durch den Pseudobefehl .en.

Sollte während der Assemblierung ein Fehler entdeckt werden und das Objektfile nicht schon durch die HYPRA-ASS-Fehlerroutine geschlossen worden sein, geben Sie bitte CLOSE 14 ein.

Ausgabe von formatierten Listings

1) **.li 1,3,0** sendet ein formatiertes Listing des Quelltextes unter der logischen Filenummer 1 an das Gerät 3 mit der Sekundäradresse 0 (Bildschirm). Die Parameter hinter .li entsprechen denen des OPEN-Befehls. So ist es auch möglich, mit .li 2,8,2, "test,u,w" das Listing auf eine Userdatei zu leiten und so weiter.

Der .li-Pseudobefehl muß der erste Befehl im Quelltext sein, wenn alle Zeilen gelistet werden sollen. Die Zeilen bis einschließlich .li werden nicht ausgegeben. Die gelisteten Zeilen haben folgendes Format:

c000 a0b0c0: 1000 -marke befehl ;kommentar

Die Steuerung der Formatierung erfolgt mit dem Editorbefehl /t. Bei Zeilen, die Pseudobefehle enthalten, wie .eq... werden keine Adressen und Opcodes ausgegeben.

2) **.sy 1,3,0** sendet am Ende von Pass 2 die sortierte Symboltabelle. Die Formatierung wird hier durch /t3,... gesteuert. Die Labelwerte werden hexadezimal ausgegeben.

Eine Zeile der Symboltabelle sieht dann folgendermaßen aus:

sprungziel = $ffd2

Das Listing des Quelltextes erhält die Kopfzeile »HYPRA-ASS Assemblerlisting:«. Die Symboltabelle erhält die Kopfzeile »Symbols in alphabetical order«.

3) **.dp t0,t1,t2,t3** setzt aus dem Quelltext heraus die Tabulatoren
t0 = Tabulator für Assemblerbefehle
t1 = Tabulator für den Kommentar bei der formatierten Ausgabe
t2 = Tabulator für die Anzahl der Blanks, die am Anfang einer Ausgabezeile ausgegeben werden.
t3 = Tabulator für die Symboltabelle

4) **.st** beendet die Assemblierung

Am Ende des zweiten Passes wird immer die Meldung »end of assembly« gefolgt von der Assemblierungsdauer in Minuten, Sekunden und Zehntelsekunden ausgegeben. Dahinter folgt die Zeile »base = $XXXX last byte at $YYYY«.

Eine Zusammenfassung aller Pseudobefehle finden Sie in Bild 3.

HYPRA-ASS-Editor

Eines der Ziele bei der Entwicklung von HYPRA-ASS war es auch,die Editierung von Quelltexten möglichst bequem zu machen. Dazu wurden etliche Funktionen, die im normalen Basic-Editor stets gebraucht, aber nie vorhanden sind, in den HYPRA-ASS-Editor eingebaut.

Als Grundlage des HYPRA-ASS-Editors blieb dabei der Basic-Editor erhalten.

Ein HYPRA-ASS-Quelltext wird also im Prinzip genauso eingegeben wie ein Basic-Programm. **Allerdings muß hinter der Zeilennummer immer ein Minuszeichen eingegeben werden,** das den Beginn der Quelltextzeile bildet. So eingegebene Quelltextzeilen werden als ASCII-Zeilen in den Speicher übernommen. Alle überflüssigen Blanks werden entfernt.

Jede eingegebene Zeile wird sofort nach der Übernahme formatiert ausgegeben, um die Übersichtlichkeit des Quelltextes zu gewährleisten. Eine Tabelle aller Editorbefehle finden Sie in Bild 1.

(Gerd Möllmann/ah)

1)	.ba $C000	gibt die Startadresse der Assemblierung an. Bei anderen Assemblern heißt dieser Befehl auch org oder * = .
2)	.eq label=wert	weist einem Label einen Wert zu
3)	.gl label=wert	weist einem globalen Label einen Wert zu
4)	.by 1,2,"a"	Einfügen von Byte-Werten in den Quelltext
5)	.wo 1234,label	Einfügen von Adressen in der Folge low/high
6)	.tx"text"	Einfügen von Text als ASCII-Werte
7)	.ap "file"	Verketten von Quelltexten
8)	.ob "file,p,w"	Senden des Objektcodes zur Floppy
9)	.en	Schließen des Objektfiles
10)	.on ausdruck,sprung	bedingter Sprung, wenn Ausdruck wahr
11)	.go sprung	unbedingter Sprung
12)	.if ausdruck	Fortführung der Assemblierung bei ELSE, falls Ausdruck falsch. Ansonsten hinter .if bis zu ELSE oder ENDIF.
13)	.el	Alternative zu den Zeilen, die hinter .if stehen
14)	.ei	Ende der IF-Konstruktion
15)	.co var1,var2	Übergabe von Labeln und Quelltext an nachgeladene Teile
16)	.ma makro (par1,par2)	Makrodefinitionszeile
17)	.rt	Ende der Makrodefinition
18)	...makro (par1,par2)	Makroaufruf
19)	.li lfn, dn, ba	sendet formatiertes Listing unter der File-Nummer lfn zum Gerät dn mit der Sekundäradresse ba
20)	.sy lfn, dn, ba	sendet formatierte Symboltabelle unter der File-Nummer lfn zum Gerät dn mit der Sekundäradresse ba
21)	.st	beendet die Assemblierung
22)	.dp t0, t1, t2, t3	setzt die Tabulatoren T0, T1, T2, T3 aus dem Quelltext heraus

Vor den Anweisungen 12, 13, 14, 16 und 17 dürfen in derselben Zeile keine Label stehen.

Bild 3. Zusammenfassung aller Pseudobefehle

$0000	Zeropage
$033e	Bandpuffer als Zwischenspeicher
$0400	Video-RAM
$0800	Hypra-ASS
$1fd7	Raum für Quelltext und Label. Quelltext bis maximal $a000
$a000	Basic-Interpreter — darunter von c000 abwärts die Symboltabelle
$c000	frei!
$d000	I/O und so weiter
$c000	Kernal

Bild 4. Speicherbelegung von HYPRA-ASS



```
programm : hypra-ass      0801 1fd7
-----------------------------------
0801 : 76 08 00 00 9e 32 31 36   28
0809 : 38 3a 22 0d 91 0e 05 20   66
0811 : 20 27 48 59 50 52 41 2d   f9
0819 : 41 53 53 20 20 20 20 20   a1
0821 : 4d 41 4b 52 4f 41 53 53   1f
0829 : 45 4d 42 4c 45 52 0d 20   8a
0831 : 20 c0 c0 c0 c0 c0 c0 c0   90
0839 : c0 c0 c0 c0 c0 c0 c0 c0   38
0841 : c0 c0 c0 c0 c0 c0 c0 c0   40
0849 : c0 c0 c0 c0 c0 c0 0d 20   38
0851 : 20 20 56 4f 4e 0d 0d 20   c3
0859 : 20 20 c7 45 52 44 20 cd   87
0861 : 4f 45 4c 4c 4d 41 4e 4e   a4
0869 : 20 20 20 31 39 38 35 20   32
0871 : 28 c3 29 0d 00 00 00 4c   ff
0879 : 55 0b 43 50 58 43 50 59   c2
0881 : 4c 44 58 4c 44 59 43 4d   46
0889 : 50 41 44 43 41 4e 44 44   13
0891 : 45 43 45 4f 52 49 4e 43   e2
0899 : 4c 44 41 41 53 4c 42 49   b3
08a1 : 54 4c 53 52 4f 52 41 52   6c
08a9 : 4f 4c 52 4f 52 53 42 43   ec
08b1 : 53 54 41 53 54 58 53 54   e7
08b9 : 59 4a 4d 50 4a 53 52 54   c6
08c1 : 58 41 54 41 58 54 59 41   07
08c9 : 54 41 59 54 53 58 54 58   99
08d1 : 53 50 48 50 50 4c 50 50   b2
08d9 : 48 41 50 4c 41 42 52 4b   65
08e1 : 52 54 49 52 54 53 4e 4f   b2
08e9 : 50 43 4c 43 53 45 43 43   49
08f1 : 4c 49 53 45 49 43 4c 56   ec
08f9 : 43 4c 44 53 45 44 44 45   f0
0901 : 59 49 4e 59 44 45 58 49   20
0909 : 4e 58 42 50 4c 42 4d 49   bd
0911 : 42 56 43 42 56 53 42 43   27
0919 : 43 42 43 53 42 4e 45 42   e9
0921 : 45 51 e4 c4 a6 a4 c5 65   52
0929 : 25 c6 45 e6 a5 06 24 46   87
0931 : 05 26 66 e5 85 86 84 4c   d7
0939 : 20 8a aa 98 a8 ba 9a 08   37
0941 : 28 48 68 00 40 60 ea 18   8a
0949 : 38 58 78 b8 d8 f8 88 c8   eb
0951 : ca e8 10 30 50 70 90 b0   c6
0959 : d0 f0 40 40 54 68 7b 7b   27
0961 : 7b 28 7b 28 7b a8 00 a8   23
0969 : 7b a8 a8 7b 3b 04 08 00   c6
0971 : 00 00 00 00 00 02 02 02   8e
0979 : 02 03 03 02 01 0c fc 10   83
0981 : 10 14 18 04 04 2b 2d 2a   c5
0989 : 2f 5e 41 4f 3e 3d 3c 43   67
0991 : 41 4e 27 54 20 4e 55 4d   b2
0999 : 42 45 52 20 54 45 52 4d   6a
09a1 : 00 4e 4f 20 4d 4e 45 4d   97
09a9 : 4f 4e 49 43 00 49 4c 4c   ee
09b1 : 45 47 41 4c 20 52 45 47   ac
09b9 : 49 53 54 45 52 00 45 4e   40
09c1 : 44 20 4f 46 20 4c 49 4e   d8
09c9 : 45 20 45 58 53 50 45 43   ce
09d1 : 54 45 44 00 57 52 4f 4e   bb
09d9 : 47 20 41 44 44 52 45 53   9c
09e1 : 53 00 42 52 41 4e 43 48   33
09e9 : 20 54 4f 4f 20 46 41 52   cf
09f1 : 00 55 4e 4b 4e 4f 57 4e   f2
09f9 : 20 4c 41 42 45 4c 00 49   21
0a01 : 4c 4c 45 47 41 4c 20 4c   3d
0a09 : 41 42 45 4c 00 54 4f 4f   c5
0a11 : 20 4d 41 4e 59 20 4c 41   3c
0a19 : 42 45 4c 53 00 4e 4f 20   6b
0a21 : 4d 41 43 52 4f 20 54 4f   10
0a29 : 20 43 4c 4f 53 45 00 55   f2
0a31 : 4e 4b 4e 4f 57 4e 20 50   ab
0a39 : 53 45 55 44 4f 00 4c 41   b5
0a41 : 42 45 4c 20 44 45 43 4c   51
0a49 : 41 52 45 44 20 54 57 49   22
0a51 : 43 45 00 50 41 52 41 4d   87
0a59 : 45 54 45 52 00 52 45 54   b4
0a61 : 55 52 4e 00 90 09 a2 09   61
0a69 : ae 09 bf 09 d5 09 e3 09   f4
0a71 : f2 09 00 0a 0e 0a 1e 0a   e7
0a79 : 30 0a 3f 0a 54 0a 0a aa   d3
0a81 : bd 65 0a 85 22 bd 66 0a   e2
0a89 : 85 23 a9 0e 20 c3 ff 20   2c
0a91 : cc ff a9 00 85 13 20 d7   e8
0a99 : aa a0 00 b1 22 f0 06 20   cc
0aa1 : d2 ff c8 d0 f6 4c 62 a4   64
0aa9 : a9 00 85 0d 20 73 00 b0   54
0ab1 : 06 20 f3 bc 4c 0c 0b 20   ed
0ab9 : 13 b1 90 06 20 b1 0d 4c   e6
0ac1 : 0c 0b 20 79 00 10 03 4c   af
0ac9 : b1 ae c9 2b f0 de c9 24   1e
0ad1 : f0 20 c9 2d f0 16 c9 22   14
0ad9 : f0 15 c9 21 f0 17 c9 3e   56
0ae1 : f0 5a c9 3c f0 56 20 f1   1e
0ae9 : ae 4c 0c 0b 4c 0d af 4c   a6
0af1 : bd ae 4c 6a 0d 20 73 00   05
0af9 : c9 4e d0 0a 20 73 00 c9   90
0b01 : 21 d0 03 4c d0 ae a9 00   fe
0b09 : 4c 7f 0a 20 79 00 aa 10   fe
0b11 : 01 60 68 68 8a c9 21 d0   87
0b19 : 0c 20 73 00 aa 20 73 00   8c
0b21 : c9 21 d0 e2 8a a2 0a dd   ad
0b29 : 85 09 f0 05 ca d0 f8 f0   09
0b31 : 04 8a 18 69 a9 a2 00 86   6a
0b39 : 4d 4c bb ad 48 20 73 00   a4
0b41 : 20 f1 ae 20 f7 b7 aa 68   c2
0b49 : c9 3c f0 02 8a a8 20 a2   60
0b51 : b3 4c 0c 0b a9 9d a2 17   cf
0b59 : 8d 02 03 8e 03 03 a9 a9   bc
0b61 : a2 0a 8d 0a 03 8e 0b 03   84
0b69 : a9 47 a2 0f 8d 08 03 8e   83
0b71 : 09 03 a9 1f 85 2c a9 d8   5c
0b79 : 85 2b a9 00 8d d7 1f 20   53
0b81 : 44 a6 a2 fa 9a 4c 49 a8   a3
0b89 : a2 00 86 41 a0 00 b1 7a   bb
0b91 : dd 7b 08 d0 10 c8 b1 7a   4b
0b99 : dd 7c 08 d0 08 c8 b1 7a   53
0ba1 : dd 7d 08 f0 0e e6 41 e8   4c
0ba9 : e8 e8 e0 a8 d0 de a9 01   ff
0bb1 : 4c 7f 0a a6 41 bd 23 09   b5
0bb9 : 85 3b a9 02 4c fc a8 a9   29
0bc1 : 00 85 3e 20 73 00 c9 23   bc
0bc9 : f0 4f c9 28 d0 55 20 2a   65
0bd1 : 14 a5 3d d0 14 20 79 00   49
0bd9 : c9 2c f0 22 c9 29 d0 19   94
0be1 : 20 73 00 c9 2c f0 27 d0   7d
0be9 : 03 20 73 00 a5 41 c9 15   8f
0bf1 : d0 07 a9 09 85 3e 4c 6d   27
0bf9 : 0c a9 04 4c 7f [illegible] 20 73   14
0c01 : 00 a9 58 20 ff ae 20 f7   d6
0c09 : ae e6 3e 4c 6d 0c 20 73   e2
0c11 : 00 a9 59 20 ff ae 4c 6d   c2
0c19 : 0c 20 38 14 a9 06 85 3e   23
0c21 : 4c 6d 0c a9 07 85 3e 20   32
0c29 : 79 00 f0 3b c9 3b f0 37   ee
0c31 : a2 05 86 3e 20 2d 14 20   bb
0c39 : 79 00 c9 2c d0 22 20 73   30
0c41 : 00 aa 20 73 00 8a c9 58   39
0c49 : f0 0b c9 59 f0 05 a9 02   3e
0c51 : 4c 7f 0a c6 3e a5 3d d0   60
0c59 : 04 c6 3e c6 3e 4c 6d 0c   3d
0c61 : a9 0a 85 3e 4c 56 0c a9   33
0c69 : 07 85 3e 60 20 79 00 f0   7e
0c71 : fa c9 3b f0 f6 a9 03 4c   9e
0c79 : 7f 0a a6 3e e0 06 d0 0c   08
0c81 : a5 41 c9 04 b0 06 a5 3b   02
0c89 : e9 07 85 3b e0 09 f0 29   2b
0c91 : a5 41 c9 15 90 35 f0 14   8a
0c99 : c9 16 f0 10 c9 30 90 03   12
0ca1 : 4c 28 0d e0 07 d0 1f a9   27
0ca9 : 01 85 42 60 e0 08 f0 04   24
0cb1 : e0 0a d0 12 a9 03 85 42   5a
0cb9 : 60 a5 41 c9 15 d0 07 a9   bd
0cc1 : 6c 85 3b 4c b5 0c a9 04   b3
0cc9 : 4c 7f 0a a9 02 85 42 e0   a4
0cd1 : 08 f0 0d e0 0a d0 0a e6   ce
0cd9 : 42 a5 3b 18 69 08 85 3b   23
0ce1 : 60 a9 01 e0 00 f0 05 0a   22
0ce9 : ca 4c e4 0c a6 41 3d 5b   b4
0cf1 : 09 d0 11 a6 3e e0 02 f0   50
0cf9 : 04 e0 03 d0 c9 e8 e8 86   dd
0d01 : 3e 4c e2 0c a6 3e e0 04   87
0d09 : d0 0f a5 41 c9 02 d0 09   f4
0d11 : a9 03 85 42 a9 be 85 3b   03
0d19 : 60 bd 76 09 85 42 bd 7e   75
0d21 : 09 18 65 3b 85 3b 60 20   eb
0d29 : ad 0c c6 42 a5 fd c9 02   4c
0d31 : 90 f4 a5 3c a6 3d 38 e5   2d
0d39 : fb a8 8a e5 fc 90 17 aa   ee
0d41 : 98 e9 02 b0 01 ca 85 3c   59
0d49 : e0 00 d0 05 c9 80 b0 01   63
0d51 : 60 a9 05 4c 7f 0a aa 98   75
0d59 : 38 e9 02 b0 01 ca 85 3c   11
0d61 : e0 ff d0 ed c9 80 90 e9   e9
0d69 : 60 a2 0a a9 00 95 5d ca   8a
0d71 : d0 fb 20 73 00 90 0b 20   a6
0d79 : 13 b1 90 17 e9 07 c9 40   ea
0d81 : b0 11 e9 2f 20 7e bd a5   52
0d89 : 61 18 69 04 85 61 90 e2   3d
0d91 : 4c 7e b9 a5 61 e9 03 90   d2
0d99 : 05 85 61 4c 0c 0b 4c 08   9d
0da1 : af 85 62 84 63 a2 90 38   3a
0da9 : 4c 49 bc 00 00 00 00 00   c9
0db1 : a2 00 86 0d 86 0c 86 0e   96
0db9 : a5 7a a4 7b 85 49 84 4a   7d
0dc1 : 20 79 00 20 13 b1 b0 09   36
0dc9 : a9 20 85 81 a9 07 4c 7f   17
0dd1 : 0a 20 73 00 90 05 20 13   a0
0dd9 : b1 90 04 e8 4c d2 0d c9   14
0de1 : 27 f0 f8 e8 86 45 a9 36   81
0de9 : 85 01 a9 f9 a2 bf 86 60   9b
0df1 : 85 5f e4 30 90 4a d0 04   0c
0df9 : c5 2f 90 44 a0 01 b1 5f   9a
0e01 : c9 ff f0 0e cd ad 0d d0   e8
0e09 : 2a 88 b1 5f cd ac 0d d0   e8
0e11 : 22 c8 c8 b1 5f c5 45 d0   da
0e19 : 1a c8 b1 5f 85 47 c8 b1   09
0e21 : 5f 85 48 a0 00 b1 49 d1   bf
0e29 : 47 d0 08 c8 c4 45 d0 f5   99
0e31 : 4c a5 0e a5 5f 38 e9 07   f6
0e39 : a6 60 b0 b4 ca 4c ef 0d   bb
0e41 : 68 48 c9 bf d0 19 20 9f   cd
0e49 : 0e a6 fd f0 08 ca f0 08   f2
0e51 : a9 06 4c 7f 0a a9 00 2c   47
0e59 : a9 80 a0 00 4c a2 0d a5   c4
0e61 : 5f a4 60 c4 2e 90 06 d0   e4
0e69 : 07 c5 2d b0 03 4c b7 0e   42
0e71 : 85 2f 84 30 a0 00 ad ac   cf
0e79 : 0d 91 5f c8 ad ad 0d 91   df
0e81 : 5f c8 a5 45 91 5f c8 a5   d9
0e89 : 49 91 5f c8 a5 4a 91 5f   3d
0e91 : a5 5f a4 60 18 69 05 90   1d
0e99 : 01 c8 85 49 84 4a a9 37   39
0ea1 : 85 01 38 60 20 91 0e a9   db
0ea9 : 36 85 01 20 83 14 20 9f   7f
0eb1 : 0e 20 a6 0d 18 60 20 9f   5f
0eb9 : 0e a9 08 4c 7f 0a 00 00   70
0ec1 : 00 00 a9 00 85 fd 85 fe   88
0ec9 : 4c e4 a7 a0 02 b1 7a f0   ff
0ed1 : 1f c8 b1 7a 85 39 c8 b1   b9
0ed9 : 7a 85 3a a4 7b a6 7a e8   e2
0ee1 : d0 01 c8 8e c1 0e 8c c2   7a
0ee9 : 0e a0 04 20 fb a8 18 60   73
0ef1 : 38 60 00 0f 1e 02 0f a5   b5
0ef9 : fd c9 02 90 43 ad 32 16   04
0f01 : 20 c3 ff a9 ff 85 3a a9   a0
0f09 : 00 8d ac 0d 8d ad 0d 8d   32
0f11 : af 0d a9 01 8d ae 0d ad   af
0f19 : e0 16 f0 12 20 d7 aa 20   2e
0f21 : 03 17 20 d7 aa 20 d7 aa   13
0f29 : ae db 16 20 c9 ff 20 3e   e8
0f31 : 15 20 a6 15 20 cc ff ad   66
0f39 : db 16 20 c3 ff 4c 7b e3   b8
0f41 : 20 78 14 4c e9 0f 24 9d   0f
0f49 : 10 03 4c c3 0e 20 1f 15   ef
0f51 : a2 fa 9a 20 8e a6 a5 2b   26
0f59 : a6 2c 8d 74 09 8e 75 09   f4
0f61 : a9 00 48 a2 c0 8d e0 16   99
0f69 : 8d 33 16 8d 72 09 8e 73   58
0f71 : 09 ad 72 09 ae 73 09 85   c4
0f79 : 2f 86 30 a9 00 8d c0 0e   b8
0f81 : 8d bf 0e 8d ac 0d 8d ad   e8
0f89 : 0d 8d af 0d a9 01 8d ae   21
0f91 : 0d 20 cc 0e b0 52 20 73   a8
0f99 : 00 c9 2e d0 42 20 73 00   16
0fa1 : c9 4d f0 1e c9 52 d0 37   f2
0fa9 : 20 73 00 c9 54 d0 10 ee   a6
0fb1 : c0 0e ad bf 0e cd c0 0e   4a
0fb9 : b0 25 a9 09 4c 7f 0a 4c   09
0fc1 : 08 af 20 73 00 a9 41 20   aa
0fc9 : ff ae 20 b1 0d a0 00 a5   7f
0fd1 : 7a 91 49 c8 a5 7b 91 49   8e
0fd9 : a9 ff 91 5f ee bf 0e 20   37
0fe1 : 09 a9 20 fb a8 4c 92 0f   9c
0fe9 : e6 fd a9 00 8d ac 0d 8d   c6
0ff1 : ad 0d 8d af 0d a9 01 8d   bb
0ff9 : ae 0d 20 8e a6 20 b4 16   72
```

Listing »HYPRA-ASS«. Bei der Eingabe bitte Seite 77 beachten.

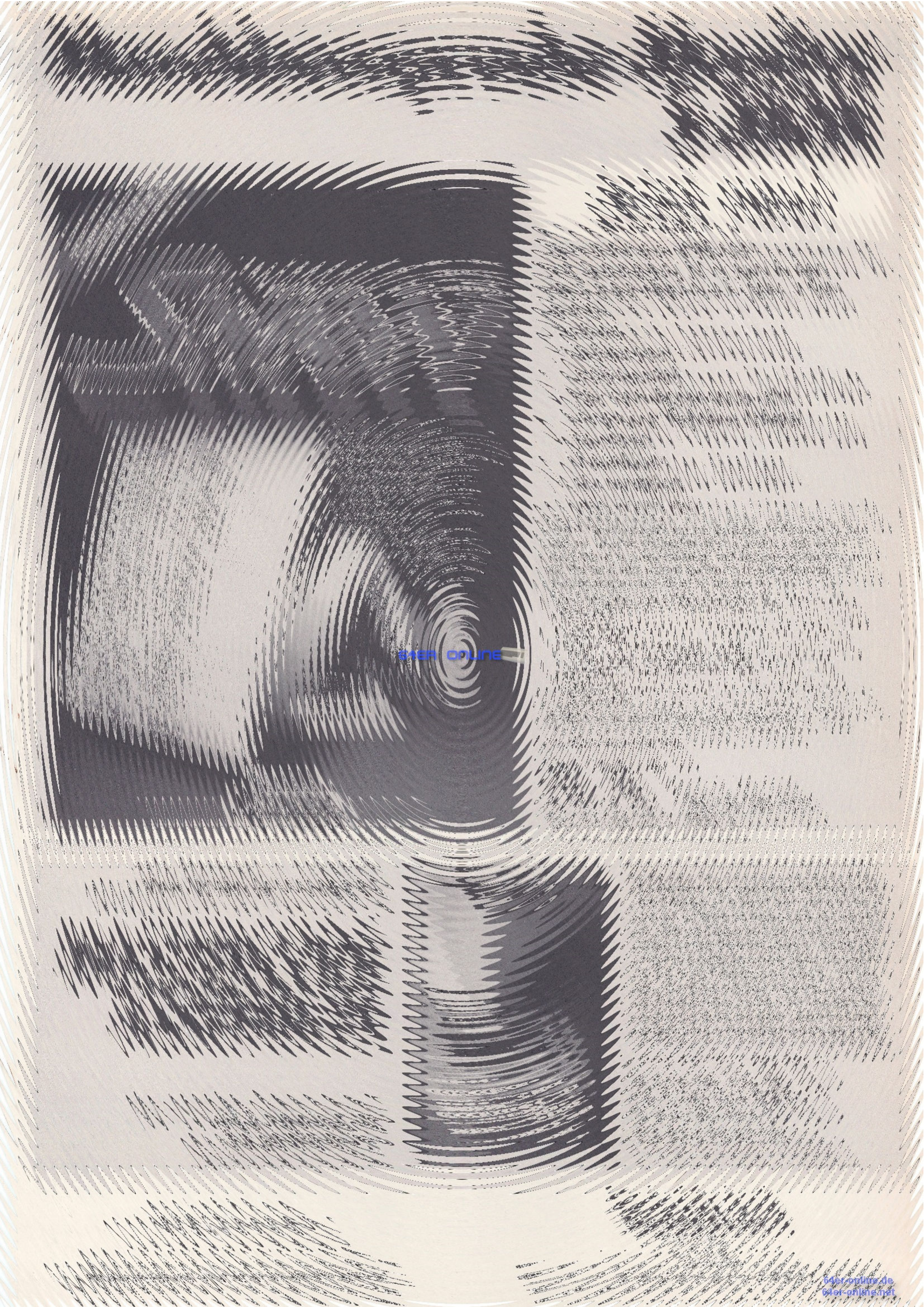


```
1001 : 20 cc 0e 90 06 4c f8 0e  e0
1009 : 4c 96 10 a0 01 b1 7a c9  d4
1011 : 3b f0 f5 c9 20 f0 4e c9  d1
1019 : 2e f0 4a a5 fd c9 01 f0  1b
1021 : 0f c8 b1 7a f0 5b c9 20  a1
1029 : d0 f7 20 fb a8 4c 69 10  2f
1031 : a9 ff 85 81 20 73 00 20  49
1039 : b1 0d a9 20 85 81 b0 15  31
1041 : a5 4a cd 73 09 90 09 d0  c8
1049 : 0c a5 49 cd 72 09 b0 05  70
1051 : a9 0b 4c 7f 0a a5 fc a6  92
1059 : fb 20 91 14 20 79 00 c9  ad
1061 : 00 d0 08 f0 1c 20 73 00  7a
1069 : 20 79 00 c9 2e f0 55 20  7f
1071 : 5f 14 a0 00 b1 7a f0 14  dd
1079 : c9 3b f0 10 a9 03 4c 7f  01
1081 : 0a ad 33 16 f0 03 20 65  64
1089 : 17 20 73 00 20 09 a9 20  be
1091 : fb a8 4c fe 0f ad 33 16  2b
1099 : f0 ef ae 32 16 20 c9 ff  fc
10a1 : 20 57 17 ad c1 0e ae c2  b5
10a9 : 0e 85 5f 86 60 a2 00 20  7e
10b1 : 82 1b 20 d7 aa 20 cc ff  a3
10b9 : 20 73 00 20 09 a9 20 fb  ed
10c1 : a8 4c 01 10 a2 00 a0 01  80
10c9 : b1 7a dd 9c 14 d0 08 c8  3c
10d1 : b1 7a dd 9d 14 f0 0b e8  b1
10d9 : e8 e0 2c d0 e9 a9 0a 4c  03
10e1 : 7f 0a bd c9 14 48 bd c8  1a
10e9 : 14 48 ad 33 16 f0 03 20  28
10f1 : 65 17 a9 02 4c fc a8 20  1c
10f9 : 73 00 20 b1 0d a9 ff a0  0a
1101 : 01 91 5f 4c 0d 11 20 73  ed
1109 : 00 20 b1 0d a9 3d 20 ff  2c
1111 : ae a5 4a 48 a5 49 48 20  33
1119 : 2d 14 68 85 49 68 85 4a  9e
1121 : a5 3d a6 3c 20 91 14 4c  0d
1129 : 73 10 20 2a 14 8d 71 09  77
1131 : 8c 70 09 20 78 14 4c 73  7c
1139 : 10 a9 01 85 42 20 38 14  3d
1141 : 85 3b 20 d3 13 20 79 00  fe
1149 : c9 2c f0 f1 4c 73 10 a9  97
1151 : 01 85 42 a0 00 20 73 00  88
1159 : c9 22 f0 03 4c 99 ad c8  aa
1161 : b1 7a f0 11 c9 22 f0 0c  37
1169 : 85 3b 98 48 20 d3 13 68  79
1171 : a8 4c 60 11 c8 20 fb a8  48
1179 : 4c 73 10 a9 02 85 42 20  4e
1181 : 2a 14 85 3c 84 3b 20 d3  e8
1189 : 13 20 79 00 c9 2c f0 ef  ac
1191 : 4c 73 10 a6 fd ca f0 2a  be
1199 : 20 e7 ff 20 73 00 20 f4  52
11a1 : 14 a9 0e a0 6e a2 08 20  7e
11a9 : ba ff 20 c0 ff a2 0e 20  10
11b1 : c9 ff a5 fb 20 d2 ff a5  46
11b9 : fc 20 d2 ff 20 cc ff 4c  7b
11c1 : 8d 10 a9 80 85 fe 4c 8d  6d
11c9 : 10 a5 fd c9 02 90 0c a9  8d
11d1 : 00 85 fe a9 0e 20 c3 ff  7a
11d9 : 20 cc ff 4c 8d 10 20 09  d5
11e1 : a9 20 fb a8 a0 04 20 fb  51
11e9 : a8 20 73 00 c9 2e d0 ee  ad
11f1 : 20 73 00 c9 52 d0 e7 4c  e8
11f9 : 8d 10 ad ac 0d 0d ad 0d  99
1201 : 8d b0 0d 20 73 00 20 b1  49
1209 : 0d a5 7b 48 8d d2 13 a5  d8
1211 : 7a 48 8d d1 13 a9 36 48  35
1219 : 85 01 a0 01 b1 49 85 7b  da
1221 : 88 b1 49 85 7a a9 37 85  62
1229 : 01 20 79 00 c9 28 d0 4e  56
1231 : 20 bc 13 20 79 00 c9 28  87
1239 : d0 4a 20 2a 14 20 bc 13  d7
1241 : 20 8a 12 20 73 00 20 b1  4a
1249 : 0d a5 3d a6 3c 20 91 14  80
1251 : 20 79 00 c9 29 f0 14 c9  65
1259 : 2c d0 29 20 bc 13 20 79  14
1261 : 00 c9 2c d0 1f 20 ae 12  3d
1269 : 4c 3b 12 20 bc 13 20 79  b3
1271 : 00 c9 29 d0 0f 20 bc 13  c5
1279 : 20 a5 12 4c 8d 10 20 8a  69
1281 : 12 4c 79 12 a9 0c 4c 7f  85
1289 : 0a ad b0 0d d0 0d ad ae  c1
1291 : 0d ae af 0d 8d ac 0d 8e  12
1299 : ad 0d 60 ee ac 0d d0 03  3f
12a1 : ee ad 0d 60 ee ae 0d d0  ef
12a9 : 03 ee af 0d 60 ad b0 0d  01
12b1 : d0 09 a9 00 8d ac 0d 8d  fe
12b9 : ad 0d 60 ad ac 0d d0 03  37
12c1 : ce ad 0d ce ac 0d 60 68  08
12c9 : c9 36 d0 1d 68 85 7a 68  f3
12d1 : 85 7b 68 48 c9 36 d0 06  d5
12d9 : 20 bc 12 4c 8d 10 a9 00  65
12e1 : 8d ac 0d 8d ad 0d 4c 8d  49
12e9 : 10 a9 0d 4c 7f 0a a6 fd  7a
12f1 : ca d0 36 20 73 00 f0 14  d8
12f9 : c9 3b f0 10 20 b1 0d 20  a2
1301 : 79 00 c9 2c d0 06 20 73  17
1309 : 00 4c fd 12 a5 2f a6 30  c0
1311 : 8d 72 09 8e 73 09 20 09  fe
1319 : a9 c8 a6 7b 18 98 65 7a  10
1321 : 90 01 e8 8d 74 09 8e 75  d2
1329 : 09 4c 8d 10 a5 fd c9 02  33
1331 : 90 f7 20 cc ff a5 fb a6  c9
1339 : fc 8d 70 09 8e 71 09 20  12
1341 : 73 00 20 f4 14 a2 08 a0  13
1349 : 00 20 ba ff a9 00 85 0a  cd
1351 : ae 74 09 ac 75 09 20 75  1c
1359 : e1 a9 00 85 fd a5 fe f0  aa
1361 : 05 a2 0e 20 c9 ff 4c 72  f2
1369 : 0f 20 73 00 20 8a ad a5  be
1371 : 61 d0 0c 20 fd ae 20 2d  72
1379 : 14 20 a3 a8 4c fe 0f 4c  2d
1381 : 8d 10 20 2a 14 4c 7a 13  17
1389 : 20 73 00 20 8a ad a5 61  d6
1391 : f0 03 4c 73 10 20 09 a9  fe
1399 : c8 c8 c8 c8 20 fb a8 20  d5
13a1 : 73 00 c9 2e d0 ef 20 73  40
13a9 : 00 c9 45 d0 e8 20 73 00  57
13b1 : c9 4c f0 04 c9 49 d0 dd  43
13b9 : 4c 8d 10 a6 7a a4 7b ad  bb
13c1 : d1 13 85 7a ad d2 13 85  95
13c9 : 7b 8e d1 13 8c d2 13 60  ce
13d1 : 00 00 a6 fd ca f0 24 a5  4b
13d9 : fe d0 2c a5 fb a6 fc 85  f3
13e1 : 22 86 23 a6 42 a0 00 a5  58
13e9 : 3b 91 22 ca f0 0d c8 a5  b5
13f1 : 3c 91 22 ca f0 05 c8 a5  7d
13f9 : 3d 91 22 a5 42 18 65 fb  af
1401 : 85 fb 90 02 e6 fc 60 a2  05
1409 : 0e 20 c9 ff a6 42 a5 3b  23
1411 : 20 d2 ff ca f0 0d a5 3c  7a
1419 : 20 d2 ff ca f0 05 a5 3d  44
1421 : 4c 19 14 20 cc ff 4c fc  fb
1429 : 13 20 73 00 20 8a ad 20  77
1431 : f7 b7 85 3d 84 3c 60 20  f9
1439 : 73 00 c9 22 f0 09 20 2d  95
1441 : 14 c9 00 d0 16 98 60 a0  3d
1449 : 01 b1 7a 85 3c c8 b1 7a  38
1451 : c9 22 d0 07 c8 20 fb a8  0f
1459 : a5 3c 60 4c 48 b2 20 89  6c
1461 : 0b 20 c0 0b 20 7b 0c a6  69
1469 : fd ca f0 08 ad 33 16 f0  b7
1471 : 03 20 4e 16 4c d3 13 ad  e6
1479 : 70 09 ae 71 09 85 fb 86  01
1481 : fc 60 a0 01 b1 49 85 62  36
1489 : 88 b1 49 85 63 84 70 60  ca
1491 : a0 01 91 49 88 8a 91 49  f5
1499 : 84 70 60 45 51 42 41 42  c7
14a1 : 59 54 58 57 4f 4f 42 45  28
14a9 : 4e 4d 41 2e 2e 52 54 43  01
14b1 : 4f 41 50 4f 4e 47 4f 49  8e
14b9 : 46 45 4c 45 49 47 4c 4c  f6
14c1 : 49 53 59 53 54 44 50 06  29
14c9 : 11 2a 11 39 11 4f 11 7b  22
14d1 : 11 93 11 c9 11 de 11 fa  6c
14d9 : 11 c7 12 ee 12 2c 13 69  d2
14e1 : 13 82 13 88 13 95 13 b8  a7
14e9 : 13 f7 10 f2 15 bc 16 3c  62
14f1 : 17 78 17 a6 7a a4 7b e8  6b
14f9 : d0 01 c8 86 bb 84 bc a0  61
1501 : 00 c8 b1 7a c9 22 d0 f9  06
1509 : 88 84 b7 60 45 4e 44 20  e5
1511 : 4f 46 20 41 53 53 45 4d  33
1519 : 42 4c 59 20 20 00 ad 0e  b1
1521 : dc 09 80 8d 0e dc ad 0f  f0
1529 : dc 29 fe 8d 0f dc a9 00  8a
1531 : 8d 0b dc 8d 0a dc 8d 09  fc
1539 : dc 8d 08 dc 60 a9 00 8d  e8
1541 : 0b dc 20 d7 aa 20 57 17  f5
1549 : a9 0d a0 15 20 1e ab ad  41
1551 : 0a dc 29 0f 18 69 30 20  c3
1559 : d2 ff a9 3a 20 d2 ff ad  d0
1561 : 09 dc aa 29 f0 4a 4a 4a  c7
1569 : 4a 18 69 30 20 d2 ff 8a  cd
1571 : 29 0f 18 69 30 20 d2 ff  a4
1579 : a9 2e 20 d2 ff ad 08 dc  e3
1581 : 29 0f 18 69 30 20 d2 ff  b4
1589 : 4c d7 aa 42 41 53 45 20  b8
1591 : 3d 20 24 00 20 20 4c 41  9e
1599 : 53 54 20 42 59 54 45 20  f4
15a1 : 41 54 20 24 00 20 57 17  26
15a9 : a9 8c a0 15 20 1e ab ad  60
15b1 : 71 09 20 37 16 ad 70 09  38
15b9 : 20 37 16 a9 95 a0 15 20  23
15c1 : 1e ab a4 fc a6 fb d0 01  0d
15c9 : 88 ca 98 20 37 16 8a 20  6f
15d1 : 37 16 4c d7 aa 48 59 50  14
15d9 : 52 41 2d 41 53 53 20 20  d0
15e1 : 41 53 53 45 4d 42 4c 45  ec
15e9 : 52 4c 49 53 54 49 4e 47  75
15f1 : 3a 00 a6 fd ca f0 37 20  e6
15f9 : 73 00 20 19 e2 a5 b8 8d  f1
1601 : 32 16 20 c0 ff b0 2c a9  e8
1609 : ff 8d 33 16 ae 32 16 20  74
1611 : c9 ff 20 d7 aa 20 d7 aa  3d
1619 : 20 57 17 a9 d6 a0 15 20  e7
1621 : 1e ab 20 d7 aa 20 d7 aa  78
1629 : 20 cc ff 20 65 17 4c 8d  0f
1631 : 10 00 00 4c f9 e0 48 4a  27
1639 : 4a 4a 4a 20 42 16 68 29  08
1641 : 0f 18 69 30 c9 3a 90 02  71
1649 : 69 06 4c d2 ff ae 32 16  8d
1651 : 20 c9 ff a2 00 ec f6 0e  09
1659 : b0 08 a9 20 20 d2 ff e8  e6
1661 : d0 f3 a5 fc 20 37 16 a5  93
1669 : fb 20 37 16 20 52 17 a6  43
1671 : 42 a5 3b 20 37 16 ca f0  8a
1679 : 10 a5 3c 20 37 16 ca f0  a0
1681 : 0b a5 3d 20 37 16 4c 90  29
1689 : 16 20 4d 17 20 4d 17 20  ef
1691 : 52 17 a9 3a 20 d2 ff 4c  52
1699 : cc ff ad c1 0e ae c2 0e  86
16a1 : 85 5f 86 60 ae 32 16 20  99
16a9 : c9 ff 20 4d 1b 20 d7 aa  8b
16b1 : 4c cc ff ad 33 16 f0 03  c7
16b9 : 4c 9b 16 60 a6 fd ca f0  cc
16c1 : 15 20 73 00 20 19 e2 a2  5f
16c9 : 05 b5 b7 9d da 16 ca 10  f4
16d1 : f8 a9 ff 8d e0 16 4c 8d  5a
16d9 : 10 00 00 00 00 00 00 00  ea
16e1 : 4c f9 e0 53 59 4d 42 4f  74
16e9 : 4c 53 20 49 4e 20 41 4c  94
16f1 : 50 48 41 42 45 54 49 43  a1
16f9 : 41 4c 20 4f 52 44 45 52  53
1701 : 3a 00 a2 05 bd da 16 95  bb
1709 : b7 ca 10 f8 20 c0 ff b0  b2
1711 : cf ae db 16 20 c9 ff 20  82
1719 : d7 aa 20 d7 aa 20 d7 aa  a9
1721 : 20 57 17 a9 e4 a0 16 20  d4
1729 : 1e ab 20 d7 aa 20 d7 aa  80
1731 : 20 7a 1e 20 90 1d 20 d7  3c
1739 : aa 4c cc ff a6 fd ca f0  a4
1741 : 08 a9 0e 20 c3 ff 4c f8  05
1749 : 0e 4c 8d 10 a9 20 20 d2  a4
1751 : ff a9 20 4c d2 ff a2 00  6e
1759 : ec f6 0e b0 06 20 52 17  33
1761 : e8 d0 f5 60 ae 32 16 20  50
1769 : c9 ff 20 57 17 a2 0d 20  20
1771 : 52 17 ca d0 fa 4c cc ff  61
1779 : 20 38 14 8d f4 0e 20 fd  a8
1781 : ae 20 3f 14 8d f5 0e 20  93
1789 : fd ae 20 3f 14 8d f6 0e  73
1791 : 20 fd ae 20 3f 14 8d f7  1a
1799 : 0e 4c 8d 10 a9 00 85 fd  df
17a1 : 85 fc a6 02 f0 0c 30 42  43
17a9 : ca ca f0 2d ca f0 2d ca  39
17b1 : f0 30 20 60 a5 86 7a 84  4f
17b9 : 7b 20 73 00 aa f0 f3 c9  b7
17c1 : 5f f0 13 c9 2f f0 18 a2  b6
17c9 : ff 86 3a 20 79 00 90 2c  d0
17d1 : 20 79 a5 4c e1 a7 4c 1e  69
17d9 : 1f 4c 9c 1a 4c 81 18 4c  53
17e1 : 5c 1e 4c 48 1c a9 00 85  83
17e9 : 02 60 a5 02 29 7f 85 02  6e
17f1 : 20 6b 18 20 4d 1b 20 d7  af
17f9 : aa 4c 9d 17 20 6b a9 a6  65
1801 : 7a e8 a0 05 bd 00 02 99  cf
1809 : fb 01 f0 49 c9 3b f0 27  73
1811 : c9 2e f0 23 c9 20 f0 05  fd
1819 : e8 c8 4c 05 18 e8 bd 00  d9
```

Listing »HYPRA-ASS« (Fortsetzung)

```
1821 : 02 f0 31 c9 20 f0 f6 c9  1a
1829 : 22 f0 18 c9 3b f0 07 c8  eb
1831 : 99 fb 01 4c 1e 18 c8 99  8b
1839 : fb 01 e8 bd 00 02 d0 f6  e8
1841 : 4c 55 18 c8 99 fb 01 e8  a6
1849 : bd 00 02 f0 07 c9 22 d0  8e
1851 : f2 4c 30 18 c8 99 fd 01  cc
1859 : 99 fb 01 c6 7b a9 ff 85  19
1861 : 7a a5 02 09 80 85 02 4c  24
1869 : a2 a4 a9 91 20 d2 ff a2  d8
1871 : 00 a9 20 20 d2 ff e8 e0  e4
1879 : 27 d0 f6 a9 00 85 d3 60  37
1881 : ad 3d 03 ae 3c 03 20 cd  5b
1889 : bd a2 04 20 2a 1f a9 2d  39
1891 : 20 d2 ff ad 3d 03 85 15  fc
1899 : ad 3c 03 85 14 18 6d 40  0e
18a1 : 03 8d 3c 03 90 03 ee 3d  32
18a9 : 03 20 60 a5 86 7a c8 84  f1
18b1 : 7b a2 00 ad 00 02 d0 06  92
18b9 : 20 e6 17 4c 9d 17 4c 03  65
18c1 : 18 e6 7a 20 eb b7 a5 14  2a
18c9 : a4 15 8d 3c 03 8c 3d 03  72
18d1 : 8e 40 03 a9 03 85 02 4c  72
18d9 : 9d 17 20 73 00 90 09 f0  03
18e1 : 07 c9 2d f0 03 4c 08 af  48
18e9 : 20 6b a9 20 13 a6 20 79  07
18f1 : 00 f0 0c c9 2d d0 ee 20  fb
18f9 : 73 00 20 6b a9 d0 e6 a5  ea
1901 : 14 05 15 d0 06 a9 ff 85  b0
1909 : 14 85 15 60 a0 01 b1 5f  c9
1911 : f0 13 c8 b1 5f aa c8 b1  c5
1919 : 5f c5 15 d0 04 e4 14 f0  54
1921 : 02 b0 03 18 24 38 60 a0  06
1929 : 00 b1 5f aa c8 b1 5f 85  d2
1931 : 60 86 5f 60 4c 48 b2 20  ca
1939 : db 18 a5 5f a6 60 85 19  2b
1941 : 86 1a a5 14 25 15 aa e8  38
1949 : f0 06 e6 14 d0 02 e6 15  5b
1951 : 20 13 a6 a5 5f a6 60 85  11
1959 : 24 86 25 38 e5 19 8a e5  2e
1961 : 1a 90 d1 a5 2d e5 24 85  8a
1969 : 5f a5 2e e5 25 85 60 18  13
1971 : a5 19 65 5f 85 2d a5 1a  75
1979 : 65 60 85 2e a0 00 b1 24  4f
1981 : 91 19 e6 19 d0 02 e6 1a  68
1989 : e6 24 d0 02 e6 25 a5 5f  e3
1991 : 38 e9 01 85 5f a5 60 e9  27
1999 : 00 85 60 10 e1 20 59 a6  48
19a1 : 20 33 a5 4c 74 a4 20 db  f2
19a9 : 18 20 2c a8 20 0d 19 b0  22
19b1 : 0c 20 4d 1b 20 d7 aa 20  30
19b9 : 28 19 4c aa 19 4c 74 a4  e5
19c1 : 20 9b b7 86 fe 20 fd ae  b4
19c9 : 20 8a ad 20 f7 b7 a6 fe  73
19d1 : 9d 61 03 98 9d 43 03 20  33
19d9 : fd ae 20 8a ad 20 f7 b7  b2
19e1 : a6 fe 9d 9d 03 98 9d 7f  8c
19e9 : 03 4c 74 a4 bd 43 03 85  d1
19f1 : 14 bd 61 03 85 15 86 fe  b6
19f9 : 20 13 a6 a6 fe bd 7f 03  03
1a01 : 85 14 bd 9d 03 85 15 60  25
1a09 : 20 9e b7 20 ed 19 20 44  1b
1a11 : e5 4c aa 19 20 9b b7 86  b5
1a19 : fe 20 fd ae 20 eb b7 8e  da
1a21 : 40 03 a5 14 a6 15 8d 3c  90
1a29 : 03 8e 3d 03 a6 fe 20 ed  e2
1a31 : 19 ad 3c 03 9d 43 03 ad  ec
1a39 : 3d 03 9d 61 03 20 0d 19  23
1a41 : b0 1d ad 3d 03 91 5f 88  de
1a49 : ad 3c 03 91 5f 18 6d 40  f4
1a51 : 03 8d 3c 03 90 03 ee 3d  e2
1a59 : 03 20 28 19 4c 3e 1a ad  14
1a61 : 3c 03 38 ed 40 03 a6 fe  9f
1a69 : 9d 7f 03 b0 03 ce 3d 03  3e
1a71 : ad 3d 03 9d 9d 03 4c 74  3d
1a79 : a4 a9 02 85 02 a6 2c a5  74
1a81 : 2b 8e 3d 03 8d 3c 03 a9  bd
1a89 : ff 85 14 85 15 20 13 a6  ed
1a91 : a5 5f a6 60 85 2b 86 2c  c0
1a99 : 4c c6 1a ae 3d 03 ad 3c  c0
1aa1 : 03 86 2c 85 2b 20 e6 17  21
1aa9 : 4c 74 a4 20 9b b7 8e 49  a1
1ab1 : 1f 4c 9d 17 20 73 00 20  1e
1ab9 : 57 e2 a0 00 ae 49 1f 4c  f4
1ac1 : ba ff a9 01 2c a9 00 85  20
1ac9 : 0a 20 b5 1a 20 6f e1 4c  32
1ad1 : 74 a4 20 b5 1a 20 59 e1  22
1ad9 : 4c 74 a4 20 9b b7 20 ed  60
1ae1 : 19 20 83 1c 20 fd ae 20  5c
1ae9 : ad 1c 20 0d 19 90 06 20  bd
1af1 : e6 17 4c 74 a4 a6 60 a5  51
1af9 : 5f 18 69 04 85 5d 90 01  c7
1b01 : e8 86 5e a9 03 85 45 a0  ac
1b09 : ff e6 45 c8 b1 5d f0 22  f4
1b11 : c4 ba f0 22 d1 bb f0 f3  59
1b19 : b1 bb c9 3f f0 ed c8 b1  07
1b21 : 5d d0 fb c4 ba f0 1e 90  4a
1b29 : 1c e6 5d d0 da e6 5e 4c  21
1b31 : 08 1b c4 ba d0 0f a5 02  6f
1b39 : c9 04 d0 03 4c 04 1c 20  2e
1b41 : 4d 1b 20 d7 aa 20 28 19  9d
1b49 : 4c eb 1a 00 a0 02 b1 5f  b1
1b51 : aa c8 b1 5f 85 62 86 63  04
1b59 : a2 90 38 20 49 bc 20 df  10
1b61 : bd 20 87 b4 20 a6 b6 8d  d4
1b69 : 4c 1b 20 24 ab ae 4c 1b  67
1b71 : e0 05 b0 08 a9 20 20 d2  c3
1b79 : ff e8 d0 f4 a9 2d 20 d2  e9
1b81 : ff a0 04 b1 5f c9 3b f0  1b
1b89 : 57 c9 2e f0 53 b1 5f f0  91
1b91 : 5b c9 20 f0 08 20 d2 ff  c4
1b99 : e8 c8 4c 8e 1b 20 d2 ff  c8
1ba1 : c8 e8 ec f4 0e b0 08 a9  91
1ba9 : 20 20 d2 ff 4c a2 1b a9  27
1bb1 : 03 85 22 b1 5f f0 35 20  c8
1bb9 : d2 ff c8 e8 c6 22 d0 f3  83
1bc1 : 20 3f ab e8 b1 5f f0 24  ab
1bc9 : c9 3b f0 07 20 d2 ff e8  b7
1bd1 : c8 d0 f1 ec f5 0e b0 08  be
1bd9 : a9 20 20 d2 ff e8 d0 f3  67
1be1 : b1 5f f0 08 20 d2 ff e8  e9
1be9 : c8 4c e1 1b 60 20 9b b7  98
1bf1 : 20 ed 19 20 83 1c 20 fd  e8
1bf9 : ae 20 95 1c a9 04 85 02  75
1c01 : 4c e5 1a a5 14 48 a5 15  bf
1c09 : 48 a2 05 a0 02 b1 5f 85  2e
1c11 : 14 c8 b1 5f 85 15 c8 c4  8f
1c19 : 45 f0 09 b1 5f 9d fb 01  24
1c21 : e8 4c 17 1c a0 00 c4 b7  05
1c29 : f0 0a b1 b8 9d fb 01 e8  31
1c31 : c8 4c 27 1c a4 ba b1 5d  0e
1c39 : 9d fb 01 f0 05 e8 c8 4c  86
1c41 : 37 1c 8a a8 4c a2 a4 20  eb
1c49 : 13 a6 20 4d 1b 20 d7 aa  c9
1c51 : 68 85 15 68 85 14 a6 fe  60
1c59 : bd 7f 03 85 14 bd 9d 03  f3
1c61 : 85 15 a5 45 18 65 b7 a8  60
1c69 : b1 5f f0 13 a6 60 98 18  68
1c71 : 65 5f 85 5d 90 01 e8 86  55
1c79 : 5e 88 84 45 4c 08 1b 4c  ef
1c81 : 46 1b a9 03 85 b9 85 bc  d5
1c89 : a9 b0 85 b8 a9 d8 85 bb  f2
1c91 : 60 4c 48 b2 20 9e ad 20  6e
1c99 : 82 b7 c9 26 b0 f3 85 b7  5e
1ca1 : a0 00 b1 22 91 b8 c8 c4  7d
1ca9 : b7 d0 f7 60 20 9e ad 20  c0
1cb1 : 82 b7 f0 dd c9 26 b0 d9  4b
1cb9 : 85 ba a0 00 b1 22 91 bb  ad
1cc1 : c8 c4 ba d0 f7 60 a9 08  ed
1cc9 : 85 ba 20 b4 ff a9 6f 85  60
1cd1 : b9 20 96 ff 20 a5 ff 20  af
1cd9 : d2 ff c9 0d d0 f6 20 ab  5b
1ce1 : ff 4c 74 a4 a9 24 8d 00  aa
1ce9 : 01 20 d7 aa a9 01 a8 a2  d0
1cf1 : 00 20 bd ff a2 08 a0 60  1e
1cf9 : 20 ba ff 20 d5 f3 a5 ba  83
1d01 : 20 b4 ff a5 b9 20 96 ff  27
1d09 : a9 00 85 90 a0 03 84 fb  52
1d11 : 20 a5 ff 85 fe a4 90 d0  ad
1d19 : 30 20 a5 ff a4 90 d0 29  27
1d21 : a4 fb 88 d0 e9 a6 fe 20  0f
1d29 : cd bd a9 20 20 d2 ff 20  1c
1d31 : a5 ff a6 90 d0 13 aa f0  c4
1d39 : 06 20 d2 ff 4c 30 1d a9  12
1d41 : 0d 20 d2 ff a0 02 4c 0f  7c
1d49 : 1d 20 42 f6 20 e6 17 4c  14
1d51 : c7 1c a9 08 a0 01 85 ba  2f
1d59 : 20 b1 ff a9 6f 85 b9 20  d1
1d61 : 93 ff b1 7a f0 07 20 a8  c9
1d69 : ff c8 4c 63 1d 20 ae ff  d9
1d71 : 4c 74 a4 a5 30 c5 2e d0  61
1d79 : 06 a5 2f c5 2d f0 0d 20  a5
1d81 : 73 00 c9 21 d0 03 20 7a  25
1d89 : 1e 20 90 1d 4c 74 a4 a2  bf
1d91 : bf a9 f9 85 45 86 46 e4  bf
1d99 : 30 90 7d d0 04 c5 2f 90  d7
1da1 : 77 20 2c a8 a9 36 85 01  ad
1da9 : a0 00 b1 45 c8 11 45 d0  2a
1db1 : 56 a0 06 b1 45 85 62 88  2a
1db9 : b1 45 85 63 88 b1 45 85  11
1dc1 : 23 88 b1 45 85 22 88 b1  2c
1dc9 : 45 85 47 a9 37 85 01 20  bc
1dd1 : d7 aa 20 57 17 a0 00 a2  ac
1dd9 : 00 b1 22 20 d2 ff c8 e8  61
1de1 : c4 47 d0 f5 e8 20 3f ab  1f
1de9 : ec f7 0e 90 f7 20 3f ab  3b
1df1 : a9 3d 20 d2 ff 20 3f ab  f1
1df9 : a9 24 20 d2 ff a5 62 20  0e
1e01 : 37 16 a5 63 20 37 16 a5  78
1e09 : 45 a6 46 38 e9 07 b0 01  d6
1e11 : ca a0 37 84 01 4c 94 1d  89
1e19 : 60 4d 56 4c 53 41 44 4e  2c
1e21 : 45 47 54 50 46 58 52 49  2c
1e29 : 4b 40 21 4f 42 55 43 1a  d7
1e31 : 1a 1a 1a 18 19 1a 19 1a  dd
1e39 : 1f 19 1a fc 1b 1c 1c 1d  48
1e41 : 1d 1f 1f 1f 1f 79 c2 c5  ee
1e49 : d2 c1 37 14 a6 ab 92 c0  e0
1e51 : db e1 ed e4 c6 52 73 2e  5e
1e59 : 49 81 b9 20 73 00 b0 03  d5
1e61 : 4c 09 1a a2 17 dd 19 1e  0e
1e69 : f0 06 ca d0 f8 4c 08 af  9a
1e71 : bd 2f 1e 48 bd 45 1e 48  66
1e79 : 60 a9 36 85 01 a5 30 48  7b
1e81 : a5 2f 48 20 e6 17 4c a4  76
1e89 : 1e a5 02 f0 0b a5 19 a6  a8
1e91 : 1a 85 2f 86 30 4c 84 1e  be
1e99 : a9 37 85 01 68 85 2f 68  a0
1ea1 : 85 30 60 a9 f9 a2 bf 85  4a
1ea9 : 19 86 1a 38 e9 07 85 1b  b6
1eb1 : b0 01 ca 86 1c e4 30 d0  b1
1eb9 : 06 90 ce c5 2f 90 ca a0  57
1ec1 : 02 b1 19 85 21 b1 1b 85  aa
1ec9 : 22 c8 b1 19 85 1d b1 1b  1d
1ed1 : 85 1f c8 b1 19 85 1e b1  e8
1ed9 : 1b 85 20 a0 00 b1 1d d1  79
1ee1 : 1f f0 04 b0 1a 90 11 c8  8b
1ee9 : c4 21 f0 04 c4 22 d0 ed  77
1ef1 : a5 21 c5 22 f0 02 b0 07  cc
1ef9 : a5 1b a6 1c 4c a8 1e a0  1d
1f01 : 06 b1 19 48 88 10 fa a0  65
1f09 : 06 b1 1b 91 19 88 10 f9  eb
1f11 : c8 68 91 1b c0 06 90 f8  46
1f19 : 84 02 4c f9 1e 20 73 00  a1
1f21 : 20 a0 aa 4c 74 a4 20 3f  31
1f29 : ab e4 d3 b0 f9 60 a9 ff  9a
1f31 : a0 01 91 2b 20 33 a5 a5  99
1f39 : 22 a6 23 18 69 01 90 01  5d
1f41 : e8 85 2d 86 2e 4c 74 a4  68
1f49 : 08 a9 1f a2 d8 20 cd bd  83
1f51 : 20 3f ab a5 2c a6 2b 20  95
1f59 : cd bd 20 3f ab a5 2e a6  e3
1f61 : 2d 20 cd bd 20 3f ab a9  c7
1f69 : 00 38 e5 2d 85 63 a9 a0  00
1f71 : e5 2e 85 62 20 d1 bd a9  f6
1f79 : 66 a0 e4 20 1e ab 4c 74  c6
1f81 : a4 20 2a 14 a5 14 85 2b  aa
1f89 : a5 15 85 2c 20 44 a6 4c  f7
1f91 : 74 a4 20 9b b7 c9 2c d0  ef
1f99 : 16 e0 04 b0 12 86 21 20  51
1fa1 : 9b b7 a4 21 c0 02 f0 0a  59
1fa9 : 8a 99 f4 0e 4c 74 a4 4c  92
1fb1 : 48 b2 8a 8d f6 0e 4c 74  a1
1fb9 : a4 a9 00 85 21 20 73 00  c3
1fc1 : c9 52 f0 06 c9 48 d0 e7  a2
1fc9 : e6 21 20 9b b7 8a a4 21  60
1fd1 : 99 20 d0 4c 74 a4 ff ff  a4
```

Listing »HYPRA-ASS« (Schluß). Bei der Eingabe bitte Seite 77 beachten



64ER ONLINE

Basic-Start-Generator

Maschinenroutinen laden wie Basic-Programme und mit RUN starten, bleibt nicht länger ein Wunschtraum. Dieses Programm macht es möglich.

Wie war noch die Startadresse für den Monitor? Sie haben bestimmt auch schon öfter die Einsprungadresse eines Maschinenprogramms vergessen, wenn Sie viel mit Assembler-Programmen arbeiten. Abhilfe schafft diese kleine Routine, die Sie mit dem MSE eingeben müssen.

Das Programm macht aus einem im Speicher befindlichen Maschinenprogramm ein Programm, das einen normalen Basic-Start besitzt. Es braucht lediglich noch auf den Datenträger gespeichert zu werden — fertig! Das speichern geschieht mit SAVE"name",8(1). Damit entfällt die leidige Frage nach dem SYS-Befehl, was bei einer großen Programmsammlung schon lästig sein kann. Das so präparierte Programm kann normal (mit »,8« oder »,1«) geladen und mit RUN gestartet werden.

Ausgenommen sind Programme, die im Bereich von $0800 bis $09af und unter dem Basic- und Kernal-ROM liegen, da hier entweder der Basic-Start-Generator oder von ihm verwendete Routinen liegen.

Tips zur Bedienung

1. Laden des zu bearbeitenden Maschinenprogramms
2. NEW eingeben
3. Laden des Basic-Start-Generators mit LOAD"name",8
4. Starten mit RUN
5. Eingabe von SYS 2089,[anfang],[ende+1],[einsprung]
[anfang]: Beginn des Maschinenprogramms im Speicher
[ende]: Ende des Maschinenprogramms im Speicher
[einsprung]: Einsprungadresse, also der SYS-Wert, mit dem normalerweise das Programm gestartet wurde.

Alle Parameter müssen in dezimaler Schreibweise eingegeben werden, wobei der korrekte Bereich (0 bis 65535) selbst überprüft werden muß. Es wird ein »ILLEGAL QUANTITY ERROR« ausgegeben, falls [anfang] > = [ende] ist.

6. Speichern des erzeugten Programms mit SAVE "name",8

Funktion

Nach der Eingabe und Umrechnung der Parameter in das 16-Bit-Format wird das Quellprogramm direkt hinter den Start-Generator kopiert (ab $08af). Dem Quellprogramm ist nun sein späterer Programmkopf vorgelagert. Des weiteren wird nun ein Teil des Start-Generators nach $C000 kopiert, der die weitere Arbeit übernimmt.

Jetzt werden der Programmkopf und das dahinter liegende Quellprogramm nach $0800, also dem Basic-Start, kopiert. Anschließend werden die Basic-Pointer (Anfang/Ende des Basic-Programms) und im Programmkopf die Adressen gesetzt. Nach der Meldung »READY TO SAVE« kann das bearbeitete Programm abgespeichert werden.

Nun eine nähere Erläuterung zum Programmkopf: Im Basic-ROM gibt es eine Unterroutine, die Speicherinhalte verschieben kann und deren Einsprungpunkt bei $A3BF liegt.

Diese Verschieberoutine macht sich der Basic-Start-Generator im Programmkopf zunutze, so daß das Quellprogramm wieder in seinen Ursprungsbereich kopiert werden kann. Die Übergabe der Kopierparameter erfolgt in den Adressen:
$5F/$60 — alter Blockanfang
$5A/$5B — altes Blockende + 1
$58/$59 — neues Blockende + 1

Nach dem Start des mit einem Basic-Start versehenen Programms wird die Blockverschieberoutine aktiviert und das Programm in seinen Bereich kopiert und gestartet.

Soll das Programm nach RUN nur an seinen ursprünglichen Start verschoben werden, müssen Sie als Einsprungadresse eine Adresse wählen, bei der ein RTS (return to subroutine) steht. Beispielsweise 64922, eine Adresse im Betriebssystem des C 64. RTS entspricht einem RETURN in Basic. Es bewirkt, daß das Programm zwar verschoben, aber nicht gestartet wird. Haben Sie zum Beispiel den SMON mit dem Start-Generator behandelt, meldet sich der SMON nach RUN mit der Registeranzeige, wenn Sie als Einsprungadresse 49152 angegeben haben. Haben Sie 64922 angegeben, kommt die READY-Meldung. Mit SYS49152 können Sie dann den SMON starten.

(Hermann-Josef Rottkemper/hm)

```
0990   a9   2c        LDA #2C    > Blockanfang ($082C)
0992   a0   08        LDY #08
0994   85   5f        STA 5F     > Übergabeadresse
0996   84   60        STY 60
0998   a9   ff        LDA #??    > Blockende+1
                                   (wird gesetzt)
099A   a0   ff        LDY #??
099C   85   5a        STA 5A     > Übergabeadresse
099E   84   5b        STY 5B
09A0   a9   ff        LDA #??    > neues Blockende+1
                                   (wird gesetzt)
09A2   a0   ff        LDY #??
09A4   85   58        STA 58     > Übergabeadresse
09A6   84   59        STY 59
09A8   20   bf   a3   JSR A3BF   > Sprung in Verschieberoutine
09AA   4C   ff   ff   JMP FFFF   > Einsprungadresse
                                   (wird gesetzt)
```

Assemblerlisting des Programmkopfes

```
programm : startgenerator   0801 09b0
---------------------------------------
0801 : 0b 08 0a 00 9e 32 30 36  3c
0809 : 31 00 00 00 a9 37 85 01  a7
0811 : a9 09 8d 20 d0 8d 21 d0  46
0819 : a2 00 bd a2 08 20 d2 ff  4c
0821 : e8 e0 57 d0 f5 4c 74 a4  46
0829 : 20 90 08 86 fa 85 fb 20  70
0831 : 90 08 86 fc 85 fd 20 90  f0
0839 : 08 86 fe 85 ff a5 fb c5  9d
0841 : fd 90 0b d0 06 a5 fa c5  68
0849 : fc 90 03 4c 48 b2 a9 af  f8
0851 : 85 39 a9 09 85 3a a5 fa  b5
0859 : 85 3b a5 fb 85 3c 20 f9  13
0861 : 08 a5 fc 85 2b a5 fd 85  0e
0869 : 2c a5 39 85 fa a5 3a 85  38
0871 : fb a9 00 85 39 a9 c0 85  e1
0879 : 3a a9 f9 85 3b a9 08 85  e3
0881 : 3c a9 83 85 fc a9 09 85  70
0889 : fd 20 f9 08 4c 1f c0 a9  2a
0891 : 00 85 0d 20 fd ae 20 9e  ae
0899 : ad 20 9b bc a5 64 a6 65  b8
08a1 : 60 0e 12 93 9e 20 20 20  ab
08a9 : c2 c1 d3 d4 c1 d2 d4 20  21
08b1 : 20 20 20 2d 20 20 20 31  75
08b9 : 39 38 35 20 c8 2d ca 20  c1
08c1 : d2 4f 54 54 4b 45 4d 50  8f
08c9 : 45 52 20 20 20 92 11 11  40
08d1 : 20 d3 59 4e 54 41 58 3a  20
08d9 : 20 53 59 53 32 30 38 39  5c
08e1 : 2c c1 4e 46 41 4e 47 2c  46
08e9 : c5 4e 44 45 2b 31 2c ca  12
08f1 : 4d 50 2d c9 4e 11 11 99  d0
08f9 : a2 00 a1 3b 81 39 e6 39  5b
0901 : d0 02 e6 3a e6 3b d0 02  63
0909 : e6 3c a5 3b c5 fc d0 e8  37
0911 : a5 3c c5 fd d0 e2 60 a9  fe
0919 : 83 85 3b a9 09 85 3c a5  5c
0921 : fa 85 fc a5 fb 85 fd a9  09
0929 : 00 85 39 a9 08 85 3a 20  45
0931 : 00 c0 a5 39 85 2d a5 3a  ef
0939 : 85 2e a5 2d 8d 16 08 a5  d9
0941 : 2e 8d 18 08 a5 2b 8d 1e  63
0949 : 08 a5 2c 8d 20 08 a5 fe  b7
0951 : 8d 29 08 a5 ff 8d 2a 08  4f
0959 : a2 00 bd 78 c0 20 d2 ff  d2
0961 : e8 e0 12 d0 f5 a9 01 85  14
0969 : 2b a9 08 85 2c 4c 74 a4  5c
0971 : 11 11 12 52 45 41 44 59  fc
0979 : 20 54 4f 20 53 41 56 45  be
0981 : 92 11 00 0b 08 c1 07 9e  e5
0989 : 32 30 36 31 00 00 00 a9  da
0991 : 2c a0 08 85 5f 84 60 a9  af
0999 : ff a0 ff 85 5a 84 5b a9  23
09a1 : ff a0 ff 85 58 84 59 20  f0
09a9 : bf a3 4c ff ff 00 00 20  8d
```

Listing zum Basic-Start-Generator. Beachten Sie bitte die Eingabehinweise auf Seite 77.

REM-Killer

Überflüssige REM-Anweisungen in Basic-Programmen lassen sich leicht entfernen. Dieses Programm rechnet aber zusätzlich alle Sprünge auf gelöschte REM-Zeilen um. Und das in wenigen Sekunden.

REM-Statements sind, vor allem für längere Basic-Programme, unumgänglich, um im relativ unübersichtlichen »Basic-Dschungel« den Überblick zu behalten. Spätestens dann, wenn an einem umfangreichen Programm nach einiger Zeit Änderungen erforderlich sind, wird auch der erfahrene Programmierer dankbar sein, wenn er seine Unterprogramme und undurchsichtigen PEEKs, POKEs und SYS-Befehle mit Hilfe von REMs noch durchschauen kann. Kurz gesagt, auch in sauber strukturierten Programmen gehört eine klare Dokumentation einfach dazu.

Das Problem mit den REMs

Leider haben REM-Statements die unangenehme Eigenschaft, Programme wesentlich zu verlängern und auch zu verlangsamen. Im Gegensatz zu einem Befehl (zum Beispiel POKE), werden REMarks nicht in einem Byte abgespeichert, sondern nehmen durch buchstabenweise Abspeicherung in ASCII-Codes eine Menge Platz weg. Am besten wäre daher, Basic-Programme mit ausführlicher REM-Dokumentation zu entwickeln und auch so zu speichern. Aus diesen »Quellprogrammen« können dann die REMs entfernt werden, um kürzere und schnellere »Lauf«-Versionen zu erhalten. Sollten später einmal Änderungen erforderlich werden, nimmt man diese am »Quellprogramm« vor und generiert daraus wiederum das »Laufprogramm«.

Da es sehr umständlich ist, die REMs in Handarbeit zu entfernen, braucht man ein Programm, das einem diese Arbeit abnimmt. Wozu hat man schließlich einen Computer. Das Programm sollte sehr schnell und unkompliziert zu bedienen sein; es sollte ferner Sprünge auf Zeilen, die mit einem REM beginnen (und daher vollständig gelöscht werden) korrigieren. Genau das macht diese Maschinenroutine.

Der REM-Korrektor

Wenn der REM-Korrektor mittels MSE als Maschinenprogramm auf Kassette oder Diskette gespeichert ist, kann dieses jederzeit mit LOAD"REM-KORREKTOR",8,1 (,1,1) geladen werden. Auch dann, wenn sich bereits ein Basic-Programm im Speicher befindet. Der gefürchtete OUT OF MEMORY ERROR, der durch das Nachladen von Maschinenprogrammen und dem damit verbundenen »Verbiegen« der Basic-Zeiger auftritt, wird beim Starten des REM-Korrektors mit SYS 52000 aufgehoben. Der REM-Korrektor liegt ab Adresse 52000 im Speicher, damit die Datasetten-Anwender nicht auf TURBO TAPE verzichten müssen.

Befindet sich der REM-Korrektor einmal im Computer, kann er selbstverständlich immer wieder mit SYS 52000 aufgerufen werden. Danach werden alle REMs aus dem Basic-Programm entfernt. Steht die REM-Anweisung am Zeilenanfang, wird die ganze Zeile gelöscht. Steht sie am Zeilenende, wird nur alles nach dem REM gelöscht. Sprünge auf gelöschte REM-Zeilen werden korrigiert, auch wenn die neuen Sprungadressen aus mehr ASCII-Codes bestehen wie die ursprünglichen. GOTO 99 erfordert 3 Bytes in Basic. Wird die Zeile gelöscht, und die nächste gültige Zeile ist 12300, werden für den geänderten Befehl GOTO 12300 6 Bytes erforderlich. Ferner setzt das Programm alle Basic-Zeiger in der Zeropage auf den aktuellen Wert.

Bearbeitet werden alle Sprünge wie
GOTO
GOSUB
IF...THEN (Zeilennummer)
IF...THEN GOTO (Zeilennummmer)
IF...THEN GOSUB (Zeilennummer)
IF...GOTO (Zeilennummer)
IF...GOSUB (Zeilennummer)
ON X GOTO (Zeilennummer), (Zeilennummer),...
ON X GOSUB (Zeilennummer), (Zeilennummer),...

Einer Zeilennummer vorangehende Leerzeichen sind ebenfalls erlaubt. Bei ON X GOTO oder ON X GOSUB muß jede Zeilennummer, bis auf die letzte durch ein Komma abgetrennt sein.

ON X GOTO 200, 300, 400, 500, 600 ist erlaubt. Nicht jedoch
ON X GOTO 200, 300, 400 , 500, 600.

Während des Programmablaufs erscheinen zur Kontrolle der Korrekturen die Nummern der in die Sprungbefehle neu eingesetzten Zeilen auf dem Bildschirm.

Anschließend kann mit dem Basic-Programm ganz normal weitergearbeitet werden.

Tips zur Benutzung

1. Der REM-Korrektor ist in der Lage, durch NEW versehentlich gelöschte Basic-Programme zu regenerieren (RENEW-Funktion), wenn man vor dem Starten durch SYS 52000 den Befehl POKE 2050,1 eingibt (allerdings sind dann sämtliche REMs »raus«).

2. Zeile 0 darf keine REM-Zeile sein (0REM...). Enthält Zeile 0 einen REM-Befehl, werden keine Sprungkorrekturen ausgeführt.

Der REM-Korrektor benötigt für ein 20 KByte langes Basic-Programm etwa 1,5 Sekunden, wenn keine Sprungkorrekturen notwendig sind. Aber auch wenn zahlreiche Korrekturen durchgeführt werden müssen, bleibt die Ablaufzeit im Bereich von einigen Sekunden.

Ist kein Basic-Programm im Speicher, stoppt der REM-Korrektor sofort wieder. Sind nur REM-Zeilen vorhanden, führt er einen RESET durch. Der REM-Korrektor bleibt jedoch funktionsfähig.

Das Programm arbeitet aus Geschwindigkeitsgründen mit Hilfsspeicherbereichen in denen Hilfszeiger abgelegt sind. Da diese den RAM-Speicher unter dem ROM benutzen, wird kein Basic-Speicherplatz verbraucht.

3. Die Arbeitsweise des REM-Korrektors

Das Maschinenprogramm ist in drei Hauptteile gegliedert. Im ersten Teil wird das Basic-Programm durchsucht und für jede gültige Zeile, oder den gültigen Teil einer Zeile, die Anfangsadresse (2 Bytes) und die Zahl der Bytes (1 Byte) unter dem ROM ab Adresse $A000 abgelegt.

Die Nummern der gelöschten Zeilen (2 Bytes) sowie die Nummer der jeweils nächsten gültigen Zeile (die nicht mit einem REM beginnt), werden ab Adresse $FFFF, ebenfalls unter dem ROM, in absteigender Folge abgelegt.

Anhand der Zeiger ab $A000 werden die Zeilenteile der Basic-Bereiche neu zusammengesetzt und nach untern komprimiert. Das heißt, daß mit den gültigen Zeilen oder Zeilentei-

```
programm : remkorrektor     cb20 cf12
-------------------------------------
cb20 : ad 02 08 d0 01 60 78 a9   33
cb28 : 35 85 01 a9 00 85 f7 a9   f5
cb30 : 08 85 f8 a9 00 85 f9 a9   d5
cb38 : a0 85 fa a9 ff 85 39 a9   f3
cb40 : ff 85 3a a9 00 8d ff ff   32
cb48 : 8d fe ff 8d fd ff 8d 01   1e
cb50 : a0 a0 00 b1 f7 f0 06 20   d6
cb58 : 66 cc 4c 53 cb c8 b1 f7   5c
cb60 : d0 05 c8 b1 f7 f0 51 a5   b3
cb68 : f7 85 9e a5 f8 85 9f 20   f9
cb70 : 66 cc 20 66 cc 20 66 cc   12
cb78 : 20 66 cc 20 66 cc a0 00   52
cb80 : b1 f7 f0 10 c9 8f f0 04   50
cb88 : c8 4c 80 cb c0 00 d0 03   65
cb90 : 4c 74 cc 88 98 aa a0 00   bc
cb98 : a5 9e 91 f9 20 6d cc a5   1c
cba0 : 9f 91 f9 20 6d cc 8a 91   15
cba8 : f9 20 6d cc 18 65 f7 85   3e
cbb0 : f7 90 02 e6 f8 4c 53 cb   24
cbb8 : a5 f9 8d 3c 03 a5 fa 8d   a9
cbc0 : 3d 03 ad 01 a0 d0 07 a9   0a
cbc8 : 37 85 01 4c e2 fc a9 00   48
cbd0 : 85 f7 a9 08 85 f8 a9 00   83
cbd8 : 85 f9 a9 a0 85 fa a0 00   8b
cbe0 : b1 f9 85 9e 20 6d cc b1   c7
cbe8 : f9 85 9f 20 6d cc b1 f9   88
cbf0 : aa e8 e8 e8 e8 e8 20 6d   97
cbf8 : cc 86 02 b1 9e 91 f7 c8   a6
cc00 : c4 02 d0 f7 a0 00 20 66   50
cc08 : cc a5 f7 85 9e a5 f8 85   5b
cc10 : 9f 8a 18 65 9e 85 9e 90   59
cc18 : 02 e6 9f a5 9e 91 f7 a5   cb
cc20 : 9f 20 66 cc 91 f7 ca ca   9c
cc28 : 8a 18 65 f7 85 f7 90 02   75
cc30 : e6 f8 a5 f9 cd 3c 03 d0   a7
cc38 : a7 a5 fa cd 3d 03 d0 a0   9b
cc40 : a0 00 a9 00 91 f7 20 66   71
cc48 : cc 91 f7 20 66 cc 91 f7   e2
cc50 : 20 66 cc a5 f7 85 2d 85   f6
cc58 : 2f 85 31 a5 f8 85 2e 85   ca
cc60 : 30 85 32 4c 14 cd e6 f7   a4
cc68 : d0 02 e6 f8 60 e6 f9 d0   d9
cc70 : 02 e6 fa 60 a5 f7 85 3b   57
cc78 : a5 f8 85 3c 20 f2 cc b1   b2
cc80 : 3b 91 39 20 f2 cc 20 fd   e8
cc88 : cc b1 3b 91 39 20 fd cc   54
cc90 : 20 06 cd 20 06 cd b1 3b   37
cc98 : d0 f9 20 06 cd 20 06 cd   bf
cca0 : b1 3b d0 09 20 0d cd 20   26
cca8 : 0d cd 4c ce cc a0 03 b1   ca
ccb0 : 3b a0 00 c9 8f f0 32 20   fe
ccb8 : 06 cd 20 06 cd b1 3b 91   e8
ccc0 : 39 20 f2 cc 20 fd cc b1   e8
ccc8 : 3b 91 39 20 fd cc a9 00   0b
ccd0 : 91 39 20 fd cc a9 00 91   03
ccd8 : 39 20 fd cc a9 00 91 39   8e
cce0 : 20 0d cd 20 0d cd 4c 53   15
cce8 : cb 20 06 cd 20 06 cd 4c   01
ccf0 : 93 cc c6 3b a9 ff c5 3b   2a
ccf8 : d0 02 c6 3c 60 a5 39 d0   bc
cd00 : 02 c6 3a c6 39 60 e6 3b   75
cd08 : d0 02 e6 3c 60 e6 39 d0   de
cd10 : 02 e6 3a 60 a9 00 85 f7   c1
cd18 : 8d 3c 03 a9 08 85 f8 8d   65
cd20 : 3d 03 a0 00 84 3d 84 49   de
cd28 : a0 00 b1 f7 aa d0 24 a5   41
cd30 : f7 8d 3c 03 a5 f8 8d 3d   30
cd38 : 03 20 66 cc 20 66 cc b1   4a
cd40 : f7 f0 35 84 3d 84 49 20   eb
cd48 : 66 cc 20 66 cc 20 66 cc   ea
cd50 : 4c 28 cd e0 22 d0 06 a9   54
cd58 : 01 45 3d 85 3d a5 3d d0   93
cd60 : ec e0 89 f0 19 e0 8d f0   ed
cd68 : 15 e0 a7 f0 11 a5 49 f0   3a
cd70 : dc e0 2c d0 d8 4c 7e cd   67
cd78 : a9 37 85 01 58 60 20 66   14
cd80 : cc b1 f7 c9 20 d0 03 4c   89
cd88 : 7e cd c9 30 90 bf c9 3a   08
cd90 : b0 bb 20 3a ce 84 3e b1   db
cd98 : f7 c9 2c d0 05 85 49 4c   d3
cda0 : a6 cd a9 00 85 49 a0 00   bc
cda8 : a9 ff 85 39 a9 ff 85 3a   fe
cdb0 : b1 39 d0 17 20 fd cc b1   9d
cdb8 : 39 d0 0d 20 fd cc b1 39   20
cdc0 : d0 03 4c 28 cd 20 0d cd   d8
cdc8 : 20 0d cd b1 39 c5 48 d0   9d
cdd0 : 0c 20 fd cc b1 39 c5 47   90
cdd8 : d0 06 4c ec cd 20 fd cc   cb
cde0 : 20 fd cc 20 fd cc 20 fd   f9
cde8 : cc 4c b0 cd 20 fd cc 20   25
cdf0 : fd cc b1 39 aa 20 0d cd   62
cdf8 : b1 39 a0 37 84 01 20 cd   c1
ce00 : bd a9 2c 20 ca f1 78 a0   00
ce08 : 35 84 01 a0 00 b9 00 01   a4
ce10 : f0 04 c8 4c 0d ce 98 38   d8
ce18 : e5 3e 85 4a e6 4a a0 00   0a
ce20 : c6 4a f0 06 20 87 ce 4c   1a
ce28 : 1e ce a0 00 b9 00 01 f0   57
ce30 : 06 91 f7 c8 4c 2c ce 4c   10
ce38 : 28 cd a0 00 84 47 84 48   94
ce40 : b1 f7 38 e9 30 18 65 47   20
ce48 : 85 47 a9 00 65 48 85 48   1a
ce50 : c8 b1 f7 38 e9 30 c9 0a   51
ce58 : 90 01 60 20 61 ce 4c 40   c3
ce60 : ce a5 47 85 39 a5 48 85   70
ce68 : 3a 06 47 26 48 06 47 26   5a
ce70 : 48 06 47 26 48 06 39 26   38
ce78 : 3a 18 a5 39 65 47 85 47   84
ce80 : a5 3a 65 48 85 48 60 a5   0c
ce88 : f7 85 3b a5 f8 85 3c a0   b3
ce90 : 00 b1 2d c8 91 2d a5 2d   41
ce98 : c5 3b d0 09 a5 2e c5 3c   ab
cea0 : d0 03 4c ab ce 20 09 cf   2c
cea8 : 4c 8f ce ad 3c 03 85 3b   8e
ceb0 : ad 3d 03 85 3c a0 00 b1   9a
ceb8 : 3b f0 06 20 06 cd 4c b7   60
cec0 : ce a0 02 b1 3b d0 03 4c   74
cec8 : ef ce a0 01 b1 3b aa e8   d8
ced0 : 8a 91 3b d0 08 c8 b1 3b   10
ced8 : aa e8 8a 91 3b 20 06 cd   34
cee0 : 20 06 cd 20 06 cd 20 06   d6
cee8 : cd 20 06 cd 4c b5 ce 20   ef
cef0 : 06 cd 20 06 cd 20 06 cd   37
cef8 : a5 3b 85 2d 85 2f 85 31   8c
cf00 : a5 3c 85 2e 85 30 85 32   3f
cf08 : 60 a5 2d d0 02 c6 2e c6   3d
cf10 : 2d 60 ff ff ff ff ff ff   6d
```

Listing des REM-Korrektors. Beachten Sie bitte die Eingabehinweise auf Seite 77.

len das ursprüngliche Basic-Programm überschrieben wird. Da das neue Basic-Programm kürzer oder höchstens gleich lang ist wie das alte, kann diese Operation Zeile für Zeile im selben Speicherbereich (Basic-Bereich ab 2048) stattfinden. Damit sind die REMs entfernt!

Die Teile 1 und 2 sind sehr schnell, da der Basic-Speicherbereich jeweils nur einmal durchlaufen werden muß.

Im dritten Teil wird das Basic-Programm nach Sprungcodes durchsucht. Dabei werden Bytes, die zufällig eine Sprungadresse darstellen, sich aber innerhalb von Anführungszeichen befinden (Strings), natürlich ignoriert.

Die gefundene Adresse, also die Folge von ASCII-Codes, wird in eine 2-Byte-Hexadezimalzahl umgewandelt und der Bereich von $FFFF abwärts durchsucht, ob diese Zeilennummer mit der Nummer einer gelöschten Zeile übereinstimmt. Trifft dies zu, wird die ebenfalls abgespeicherte Hexadezimalzahl (Zeilennummer) der nächsten gültigen Zeile geholt und in ASCII-Codes umgerechnet. Ist die neue Zeilennummer, in einer Folge von ASCII-Codes ausgedrückt, länger als die alte, wird das Basic-Programm um diesen Unterschied nach hinten verschoben und die Basic-Zeiger restauriert. Dann wird die Zeilennummer eingefügt und festgestellt, ob noch eine Zeilennummer folgt (ON...GOTO). Wenn nicht, wird nach weiteren Sprungcodes gesucht. (Dipl. Ing. Wilhelm Meierhofer/hm)

Hier gibt's Clubs

Computer-User-Vereinigung Bruchsal
e. V. (C. U. V. B.)
Torsten Zimmermann,
Steinacker 12, 7520 Bruchsal,
überregionaler Club,
Clubzeitung, Software-
und Hardware-Vertrieb,
Computertage, Basic-Kurse

Computergemeinschaft 64
Michael Fuhrmann,
Eilveser Hauptstr. 34,
3057 Neustadt 1,
Programmaustausch, POKE's,
Tricks, Verbindungen zwischen
C 64-Freunden

Outlaws 64
Sudetenstr. 8,
6090 Rüsselheim 7,
Spielturniere, Clubtreffen,
Clubzeitung, Tauschbörse

Software-Club-64
Hannelore Bauerhenne,
Danziger Str. 5,
6272 Niedernhausen,
Club-Mitteilungsdiskette
zum Selbstkostenpreis

LCC — Lübecker Commodore Club,
Moltkestr, 14, 2400 Lübeck,
Clubtreffen,
Softwarebibliothek,
Basic, Logo, evtl. DFÜ,
Aufnahmeprüfung:
eigenes Programm

POKIS-User & Computerclub
Gerhard Hammer,
Postfach 2886,
2000 Hamburg 20,
Clubzeitung, Grundlagen-Kurse, Freizeitclub für
Angehörige von Mitgliedern

Sigi-Soft-Computerclub
Siegmar Bröhl,
Rosenharth 4,
5207 Ruppichteroth 1,
Clubtreffen, Club-Zeitschrift,
Softwarebibliothek

Software-Changer-Club 64 (SCC 64)
Thomas Ross,
Neukirchenerstr. 55,
4150 Krefeld 11,
Adventure programmieren,
Ideenaustausch, Clubzeitung,
Programmaustausch

Falken-User-Gruppe München
St. Bonifatiusstr. 3/III,
8000 München 90,
wöchentlicher Treff,
Programmierung von
Strategiespielen, Kurse,
Softwarebibliothek,
Öffentlichkeitsarbeit zu
neuen Medien, Diskussionen

C 64-Club
Günther Schroth, Lindenstr. 9,
7273 Ebhausen 3,
Hardware-, Software-
und Erfahrungsaustausch

NO NAME-Commodore-64-Club
Ralf Hottmeyer,
Mudersbacherstr. 81,
5000 Köln 91,
Clubtreffen: monatlich,
kein Clubbeitrag, Programm-
und Informationsaustausch

C 64-Club »New Line«
Postfach 8007, 7910 Neu-Ulm 8,
Softwaretausch, Public Domain
Software, Programmieranleitungen, Programmieraufträge
annehmen und an Mitglieder
vergeben

In der nächsten Ausgabe geht's weiter mit den Clubs.

Komfortable Ein-/Ausgabe-routine

Dieses frei verschiebbare Programm für den C 64 erlaubt die Eingabe von bis zu 255 Zeichen. Es sind nur Zeichen erlaubt, die vorher definiert wurden.

Das Programm ist 261 Bytes lang und wird direkt in den Speicher geschrieben. Da es frei verschiebbar ist, verlangt der Lader zuerst die Eingabe der Startadresse und gibt dann die Einsprungadressen der beiden Routinen aus.

Eingabe-Routine:
Syntax: SYS⟨eingabe⟩, ⟨länge⟩, ⟨zeichen⟩, ⟨variable⟩
⟨eingabe⟩ entspricht der Startadresse der Routine; ⟨länge⟩ bestimmt die maximale Anzahl der Zeichen und ⟨zeichen⟩ die Zeichen selbst, die eingegeben werden dürfen. Mit ⟨variable⟩ kann die Variable selbst bestimmt werden, unter der die Eingabe abgespeichert wird. Ein alter Eintrag wird dabei überschrieben. Bevor die Eingabe beendet werden kann, muß mindestens ein Zeichen eingegeben werden. Für ⟨länge⟩ und ⟨zeichen⟩ können auch Variablen gesetzt werden.

Ausgabe-Routine:
Syntax: SYS⟨ausgabe⟩, ⟨zeile⟩, ⟨spalte⟩, ⟨druckliste⟩
Diese Routine gleicht dem PRINT-Befehl, wobei jedoch auch der Startpunkt der Ausgabe mit ⟨zeile⟩ und ⟨spalte⟩ festgesetzt werden kann.

Beispiele:

```
SYS eingabe,12, "abcdefghijklmnopqrstuvwxyz", e1$
SYS eingabe,5, "1234567890.", z$
SYS eingabe,C%,a$,b$
SYS ausgabe,0,5,"Hamader Juergen"
SYS ausgabe,ze,sp,tab (5);a;s$;"- ";
```

(Jürgen Hamader/ah)


```
1000 REM *          HAMADER JUERGEN *     <255>
1010 REM *          ORNETSMUEHL  30 *     <145>
1020 REM *          A-4910 RIED I/I *     <003>
1030 :                                    <068>
1040 PRINT CHR$(147)TAB(12)"EINGABE-AUSGAB
     E"CHR$(17)                           <100>
1050 INPUT"STARTADRESSE ";SA              <075>
1060 FOR I= SA TO SA+260                  <178>
1070 READ A:AA=AA+A:POKE I,A:NEXT         <024>
1080 IF AA<> 39363 THEN PRINT"?CHECKSUM ER
     ROR":END                             <015>
1090 PRINT CHR$(17)CHR$(17)"EINGABE: SYS"S
     A",LAENGE,TEXT,VARIABLE"             <218>
1100 PRINT CHR$(17)"AUSGABE: SYS"SA+224",S
     PALTE,ZEILE,TEXT"                    <003>
1110 DATA 32,253,174,32,158,183,224,0,240,
     18,134,252,32,253,174,32             <163>
1120 DATA 158,173,32,143,173,32,166,182,20
     1,0,208,3,76,72,178,133              <134>
1130 DATA 167,165,34,133,168,165,35,133,16
     9,169,0,133,69,32,253,174            <008>
1140 DATA 32,139,176,32,143,173,169,0,133,
     170,133,253,133,251,133,204          <083>
1150 DATA 169,96,133,254,165,198,240,252,3
     2,180,229,201,20,240,51,201          <102>
1160 DATA 13,240,81,160,0,209,168,240,7,20
     0,196,167,208,247,240,228            <013>
1170 DATA 164,251,145,253,133,171,32,210,2
     55,230,170,230,251,165,251,197       <007>
1180 DATA 252,208,209,198,251,165,252,133,
     170,169,157,32,210,255,169,0         <196>
1190 DATA 240,194,166,251,240,190,198,170,
     166,170,232,228,252,240,7,32         <199>
1200 DATA 210,255,198,251,16,174,169,32,32
     ,210,255,169,157,32,210,255          <161>
1210 DATA 169,0,240,160,166,251,240,156,16
     9,1,133,204,165,170,197,252          <165>
1220 DATA 240,9,169,32,32,210,255,169,0,24
     0,5,165,171,32,210,255               <171>
1230 DATA 160,0,165,170,145,71,32,244,180,
     152,72,138,160,1,145,71              <229>
1240 DATA 200,104,145,71,160,0,177,253,145
     ,51,200,196,170,208,247,96           <140>
1250 DATA 32,253,174,32,158,183,224,40,144
     ,3,76,72,178,138,72,32               <223>
1260 DATA 253,174,32,158,183,224,25,176,24
     1,104,168,24,32,240,255,32           <166>
1270 DATA 253,174,76,164,170              <003>
```

Listing »Ein-/Ausgaberoutine«

Hinweise zum Abtippen unserer Listings

Wenn Sie die 64'er schon öfter gelesen haben, sind Ihnen die Programme Checksummer und MSE sicher ein Begriff. Diese Programme erleichtern Ihnen die Arbeit beim Abtippen der Listings ganz erheblich.

Checksummer und Basic-Listings

Der Checksummer ist »zuständig« für alle Basic-Programme. Er zeigt nach Eingabe einer Basic-Zeile eine Prüfsumme am Bildschirm an. Wenn diese Zahl mit der in Klammern »< >« stehenden Zahl rechts von jeder Basic-Zeile in unseren Listings nicht übereinstimmt, dürfte ein Tippfehler vorliegen. Die Zahl mit den Klammern darf nicht mit eingegeben werden.

In allen Basic-Listings finden Sie Zeichen und Wörter in geschweiften Klammern. Diese stehen als Ersatz für Grafik- und Steuerzeichen und dienen der besseren Lesbarkeit. So bedeutet zum Beispiel {CLR}, daß die SHIFT und die CLEAR/HOME-Taste, oder {13SPACE}, daß 13mal die Leertaste betätigt werden muß.

In vielen Basic-Listings finden Sie unterstrichene und überstrichene Buchstaben. Unterstrichene Buchstaben bedeuten, daß Sie die SHIFT-Taste und den entsprechenden Buchstaben, überstrichene Zeichen die Commodore-Taste und den Buchstaben drücken müssen. Auf dem Bildschirm erhalten Sie dann die entsprechenden Grafikzeichen. Sie können die Basic-Listing auch ohne den Checksummer eingeben.

MSE für Assemblerprogramme

Der MSE (Maschinensprache Editor) ist eine große Hilfe bei der Eingabe von Maschinenprogrammen. Der MSE erspart ungefähr ein Drittel Tipparbeit und gestattet keine Fehleingaben. MSE-Listings sind erkennbar an den hexadezimalen Zahlenkolonnen. Als Notbehelf können Sie diese Listings aber auch mit einem Maschinensprachemonitor eingeben.

Diese Programme (Checksummer und MSE) wurden in den 64'er Ausgaben 1/85 bis 6/85 mit einer ausführlichen Erklärung abgedruckt. Sie befinden sich aber ebenfalls auf jeder Leserservice-Diskette. Gegen Einsendung eines frankierten und an Sie selbst adressierten Briefumschlags (Größe DIN C5 = 0,80 Mark, DIN C4 = 1,10 Mark Porto) schicken wir Ihnen gerne beide Listings mit einer ausführlichen Erklärung zu. □



Centronics-Interface für jeden Bedarf

Will man einen Drucker mit Centronics-Eingang an den C 64 anschließen, kommt man um ein Interface nicht herum. Wir zeigen Ihnen mit dieser Software-Schnittstelle die billigste Lösung für ein Interface.

Zweifelsohne sind Software-Lösungen von Centronics-Schnittstellen für den C 64 (mit einfachem Kabel am User-Port) nichts unbekanntes. Bedauerlicherweise ist jedoch die Verträglichkeit mit käuflichen Programmen nicht immer gewährleistet. Viele Textverarbeitungsprogramme funktionieren damit nicht. Die Software für die Schnittstelle wird durch diese Programme leider meistens überschrieben. Dieses Problem läßt sich eigentlich nur durch das feste Einbinden der »Centronics-Software« in das Betriebssystem des C 64 lösen. Der User-Port des C 64 stellt dann eine vollwertige Centronics-Schnittstelle dar. Auf alle Besonderheiten des C 64-Zeichensatzes konnte aber aus Platzgründen keine Rücksicht genommen werden. Reverse Zeichen, Cursor-Steuerzeichen und Sekundäradressen werden, bis auf die Sekundäradresse zur Umschaltung zwischen Groß- und Kleinschrift, nicht erkannt. Dafür wird aber bei Verwendung der Sekundäradresse 7 nicht nur eine Groß-/Kleinschrift, sondern auch eine C 64/ASCII-Codewandlung durchgeführt.

Centronics hoch drei

In dieser Bauanleitung werden drei Schnittstellen vorgestellt. Zum einen ist das »C 64 Centron« eine Schnittstelle, bei der außer einer Verkürzung der Systemmeldungen nichts im Betriebssystem verloren geht (auch nicht die Kassettenroutinen). Die zweite Schnittstelle ist dem in Ausgabe 4/85 veröffentlichten Hypra-Perfekt angepaßt. Bei der dritten Schnittstelle wurde auf größte Flexibilität geachtet. Es handelt sich dabei um ein frei verschiebbares Programm, das komfortabel zu bedienen ist. Außer der dritten Version sind alle Schnittstellen nur als ROM-Version benutzbar. Das Kernal-ROM wird dazu durch ein neues ersetzt. Die dritte Version ist eine reine Software-Lösung. Das Anschlußkabel, welches das in Tabelle 1 dargestellte Verdrahtungsschema haben muß, ist aber in allen drei Fällen gleich.

Software-Schnittstelle im Betriebssystem integriert

Das größte Hindernis für die Verträglichkeit mit fertigen Programmen ergibt sich aus der Tatsache, daß der C 64 bereits eine Druckerschnittstelle eingebaut hat: den seriellen IEC-Bus. Da käufliche Programme in der Regel diesen Weg benutzen, um einen Drucker anzusprechen, muß eine nachträgliche Lösung Abhilfe schaffen.

C 64 Centron (Listing 1) arbeitet mit allen Centronics-Druckern zusammen, ohne irgendeine Funktion des C 64 zu beeinflussen. Da die Schnittstellen-Software aus den oben erwähnten Gründen im Kernal-ROM selbst liegen muß, war zunächst das Platzproblem zu überwinden. Einerseits hat Commodore fast den gesamten Raum für das Kernal benutzt, andererseits ist der C 64 aber mit ausgesprochen platzaufwendigen Einschalt- und Systemmeldungen ausgestattet. Doch welcher C 64-Besitzer weiß nicht schon nach recht kurzer Zeit, welchen Namen sein Computer hat. Dementsprechend wurde die Einschaltmeldung drastisch verkürzt. Der C 64 meldet von nun an mit

* C64 CENTRON * 38911 BYTES RAM

Weiterhin wurden die Meldungen bei Betrieb von Datasette und Floppy vereinfacht. Die neuen Meldungen sind in Tabelle 2 zusammengefaßt. Als nächste Möglichkeit bieten sich die Initialisierungsroutinen des Betriebssystems an. Hier wird vom C 64 nach dem Einschalten untersucht, ob der C 64 in Europa oder in Amerika zum Einsatz kommt. Da die amerikanische Initialisierung in Europa nicht gebraucht wird, konnten die entsprechenden Routinen einschließlich der NTSC-Zeitkonstanten für die RS232-Schnittstelle aus dem Betriebssystem entfernt werden. Damit war genug freier Platz im ROM für die Schnittstellen-Software geschaffen.

Hohe Verträglichkeit einer Schnittstelle wird durch die richtige Einbindung der Änderungen in das Original-Betriebssystem erreicht. Ziel muß es dabei sein, die neuen Routinen so spät wie möglich in den normalen Ablauf der seriellen IEC-Busverbindungen einzubinden. Hier wurde folgender Weg eingeschlagen:

Nach einem Sprung aus der LISTEN-Routine ($EDOC) wird in der neuen Routine beim Ansprechen der Geräteadresse 4 (OPEN 4,4 oder OPEN 4,4,X) ein Flag für Centronics-Ausgabe (Centflag) gesetzt. Der Drucker wird durch Senden einer »0« initialisiert. Bei anderen Geräteadressen wird ein Rücksprung in die normale LISTEN-Routine durchgeführt.

In der SECLISTEN-Routine ($EDB9) wird bei gesetztem Centflag überprüft, ob bei der File-Eröffnung die Sekundäradresse 7 eingegeben wurde. Dementsprechend wird ein weiteres Flag (Secflag) gesetzt. Diese beiden Flags werden in der eigentlichen Ausgaberoutine, die in die IECOUT-Routine ($ED40) eingebunden ist, nacheinander abgefragt. Bei nichtgesetztem Centflag erfolgt ein Rücksprung in die normale IECOUT-Routine. Ist Centflag gesetzt, wird das Secflag abgefragt und bei gesetztem Flag eine Codewandlung von C 64-Code in Standard-ASCII vorgenommen. Nach Ausgabe des ASCII-Zeichens wird wie in der seriellen Ausgabe abgefragt, ob das ausgegebene Zeichen das letzte einer Zeichenkette war. Wenn ja, erfolgt ein Rücksprung in das Normalbetriebssystem. Nach Ausgabe des letzten Zeichens werden Centflag und Secflag gelöscht. Beim Schließen des Druckfiles (CLOSE4) wird »CARRIAGE RETURN« ausgegeben, um eventuell noch im Druckpuffer befindliche Zeichen auszudrucken. Dies geschieht in der SECLISTEN-Routine.

Eine besondere Einrichtung dieser Lösung ist eine Benutzerschnittstelle. Es ist nämlich möglich, vor der eigentlichen Ausgabe eines Zeichens in eigene Routinen zu verzweigen. Und das sogar bei käuflicher Software. Wichtige Voraussetzung dafür ist lediglich, daß ein RAM-Bereich frei bleibt und nie vom Programm beschrieben wird. Dies kann häufig der Kassettenpuffer ($033C-$03FB), weniger häufig der Bereich von $C000 bis$CFFF sein. Im freien RAM-Bereich muß eine Routine stehen, die indirekt angesprochen wird und nach Ausführung in die Centronics-Software zurückspringt. So können zum Beispiel Steuerzeichen auch von Druckern exotischer Herkunft aus Steuersignalen für Standarddrucker (etwa von Commodore oder Epson) erzeugt werden. In Listing 2 ist eine Beispielroutine abgedruckt, die bei Übergabe eines Wertes

von 255 an den Drucker einen Text aus der Benutzerschnittstelle ausdruckt.

Die Bedienung der Benutzerschnittstelle läuft nach folgendem Schema ab: Bei gesetztem Secflag (also bei Sekundäradresse 7) wird ein Schlüsselbyte in Speicherstelle $DC0C mit dem Wert $DB verglichen. Ist dieser Wert dort nicht vorhanden, wird die Centronics-Ausgaberoutine mit Codewandlung fortgesetzt. Wird der Wert $DB jedoch gefunden (weil er vorher hineingeschrieben worden ist), springt der C 64 über einen Zeiger in $0336/$0337 (LByte/HByte der Startadresse der eigenen Routine) ins selbstdefinierte Programm, führt die programmierten Funktionen aus und kehrt in die Ausgaberoutine zurück.

Weitere Möglichkeiten, etwa umfangreiche Codewandlung über Tabellen etc., können dadurch selbst entwickelt werden.

Die hier vorgestellte Schnittstelle arbeitet mit allen bisher getesteten Programmen zusammen, einschließlich Textomat, Textomat plus und MAE-Assembler. Leider ist es nicht mehr möglich, gleichzeitig bei Anschluß eines Druckers an den User-Port die RS232-Schnittstelle zu betreiben. Sämtliche RS232-Routinen sind jedoch unverändert vorhanden und funktionstüchtig.

Der Basic-Lader kopiert das Kernal in den RAM-Bereich von $6000 bis $7FFF und verändert dort das Betriebssystem. Das C 64 CENTRON-Betriebssystem kann nun durch Menüwahl entweder auf Diskette abgespeichert oder mit einem EPROM-Kopiergerät in ein 8-KByte-EPROM gebrannt werden. Das EPROM wird auf einer entsprechenden Adapterplatine oder auch auf einer Betriebssystem-Umschaltplatine in den Sockel U4, anstelle des normalen Kernal-ROM, eingesetzt.

Hypra-Centronics

Da bei der Programmierung des Hypra-Perfekt bereits auf die Kassettenroutinen verzichtet wurde, war die Platzsuche im Betriebssystem weniger aufwendig. Trotzdem mußte auch hier etwas getrickst werden. Die Schnittstelle fand keinen Platz mehr. Wer bereits im Besitz eines Hypra-Perfekt-Kernals ist, sollte Listing 3 eingeben und starten. Das neue Betriebssystem mit Centronics-Schnittstelle befindet sich dann wie gewohnt ab $6000 bis $8000 im Speicher. In den REM-Zeilen am Anfang des Listings ist angegeben, welche Änderungen vorgenommen werden müssen, um an Stelle des $6000-Bereichs den $8000-Bereich als Startadresse zu erhalten. Diese Änderungen können notwendig werden, wenn das EPROM-Programmiergerät nicht mit Programmen im $6000-Bereich arbeiten kann.

Softwarelösung für jedermann

Im Gegensatz zu den beiden bisher beschriebenen Versionen arbeitet die Schnittstelle aus Listing 4 im RAM-Speicher des C 64. Dafür bietet sie aber den Vorteil, völlig frei verschiebbar zu sein. Einige Bereiche, in denen die Schnittstelle keinen Platz findet, wurden von vorneherein ausgeschlossen. Eine Fehlbedienung ist somit fast unmöglich. Nach dem Starten des Programms meldet sich ein Menü, mittels dessen Sie den Speicherbereich, in dem die Schnittstelle arbeiten soll, eingeben können. Es ist dabei belanglos, ob Sie den Speicherbereich in dezimaler oder hexadezimaler Form (zum Beispiel $C000) bestimmen. Das Programm rechnet die Adressen um und informiert Sie anschließend darüber, mit welchem SYS-Befehl (der ja vom Speicherbereich abhängt) die Schnittstelle immer wieder neu initialisiert werden kann.

Wir hoffen, Ihnen mit diesen drei Lösungen das Mittel an die Hand zu geben, mit dem Sie jeden Drucker mit Centronics-Schnittstelle an den C 64 anschließen können. Es sei nochmals darauf hingewiesen, daß Listing 1 und Listing 3, ausschließlich für die Verwendung im Kernal-ROM vorgesehen sind. Dazu ist es notwendig, den Computer zu öffnen (Achtung! Garantieverlust) und das Kernal-ROM durch ein 2764-EPROM zu ersetzen. Bitte beachten Sie, daß dieser EPROM-Typ nur mit einer Adapterplatine (Bild 1 zeigt, welche Pins von Sockel und EPROM miteinander zu verbinden sind) in den U4-Steckplatz eingesetzt werden darf. Ohne diesen Adapter wird der C 64 wahrscheinlich beschädigt.

Wer möchte, kann sein neues Betriebssystem natürlich auch im RAM unter dem ROM ($E0000 bis $FFFF) ausprobieren. In Listing 1 ist dafür eine eigene Funktion im Menü vorgesehen. Bei Listing 4 brauchen Sie sich keine Gedanken um EPROMs und Steckplätze zu machen. Sie funktioniert ohne jede Änderung mit jedem C 64. Leider muß sie jedesmal vor dem Arbeiten neu geladen werden.

(W. Forstenrath, M. Braun, A. Wängler/hm)

28poliger Sockel	24poliger Sockel
1,28,27,26	24
2	21
3	1
4	2
5	3
6	4
7	5
8	6
9	7
10	8
11	9
12	10
13	11
14,20	12
15	13
16	14
17	15
18	16
19	17
20,14	12
21	19
22	20
23	18
24	22
25	23
1,28,27,26	24

Die Pins des 28poligen Sockels sind fett gedruckt

Pin 1 eines jeden Sockels ist links von der Kerbe, Pin 24 beziehungsweise 28 ist rechts davon.

Bild 1. Das Verbindungsschema für den Adaptersockel

Drucker Pin-Nummer	Drucker Signal	Signal	User-Port Pin-Nummer
1	STROBE	PA2	M
2	DATA 1	PBO	C
3	DATA 2	PB1	D
4	DATA 3	PB2	E
5	DATA 4	PB3	F
6	DATA 5	PB4	H
7	DATA 6	PB5	J
8	DATA 7	PB6	K
9	DATA 8	PB7	L
10	ACKNLG	FLAG2	B
16	GND	GND	A

Tabelle 1: Verdrahtungsschema der Centronics-Schnittstelle am User-Port (Stecker für User-Port: TRW 251-12-50-170; für Drucker: 36poliger Amphenol-Stecker; Kabel: Flach- oder Rundkabel bis 1,5 m)

Geänderte Systemmeldungen	
bisher	geändert
searching for	sr
press play on tape	pl
press record & play on tape	rc
loading	l
saving	s
verifying	v
found	entfällt
ok	+

Tabelle 2. Durch Verkürzung der Systemmeldungen wurde bei Listing 1 Platz geschaffen

```
0 REM ********************************       <207>
1 REM *                              *       <228>
2 REM *          C64 CENTRON         *       <171>
3 REM *                              *       <230>
4 REM *              VON             *       <218>
5 REM *      WOLFGANG FASTENRATH     *       <045>
6 REM *              UND             *       <208>
7 REM *         ELMAR SCHULTE        *       <115>
8 REM *                              *       <235>
9 REM *        TEL. 02246/7132       *       <249>
10 REM *                             *       <237>
11 REM *     LOHMAR, APRIL 1985      *       <044>
12 REM *                             *       <239>
13 REM *******************************       <220>
14 POKE 55,0:POKE 56,64:RESTORE              <003>
15 OF=16384:GOSUB 1000:SYS 16384             <194>
20 OF=25641:GOSUB 1000                       <229>
30 OF=25661:GOSUB 1000                       <241>
40 OF=28861:GOSUB 1000                       <000>
50 OF=30131:GOSUB 2000                       <250>
60 OF=30140:GOSUB 2000                       <004>
70 OF=30162:GOSUB 2000                       <018>
80 OF=30168:GOSUB 2000                       <034>
90 OF=30355:GOSUB 2000                       <042>
100 OF=30489:GOSUB 2000                      <060>
110 OF=30544:GOSUB 2000                      <062>
120 OF=30748:GOSUB 2000                      <078>
130 OF=30761:GOSUB 2000                      <083>
140 OF=30781:GOSUB 2000                      <095>
150 OF=32221:GOSUB 1000                      <095>
160 OF=32450:GOSUB 1000                      <109>
170 OF=27969:GOSUB 3000                      <140>
180 OF=27940:GOSUB 3000                      <139>
190 OF=28091:GOSUB 3000                      <147>
200 OF=25695:GOSUB 1000                      <162>
210 OF=25783:GOSUB 1000                      <170>
220 OF=28894:GOSUB 1000                      <185>
300 PRINT"{CLR,6RIGHT,4DOWN}<1> ABSPEICHERN <060>
310 PRINT"{3DOWN,6RIGHT}<2> TESTOPTION       <152>
315 PRINT"{3DOWN,6RIGHT}<3> ENDE             <161>
320 PRINT"{4DOWN,6RIGHT}BITTE WAEHLEN <1/2/
    3>                                       <216>
340 GET A$:IF A$=""THEN 340                  <185>
345 IF A$="1"THEN 350                        <176>
347 IF A$="2"THEN 400                        <175>
348 IF A$="3"THEN END                        <157>
349 IF A$<>"1"OR A$<>"2"OR A$<>"3"THEN 340   <068>
350 PRINT"{CLR,DOWN,RIGHT}'C64 CENTRON' (BE
    REICH $6000 - $8000)"                    <087>
360 PRINT"{DOWN,RIGHT}WIRD AUF DISKETTE GES
    PEICHERT"                                <022>
370 SYS 16421:END                            <201>
400 POKE 16388,96:POKE 16393,224:POKE 16398
    ,0:POKE 16401,0:POKE 16416,0:SYS 16384   <214>
410 POKE 16388,160:POKE 16393,160:POKE 1639
    8,0:POKE 16401,0:POKE 16416,192:SYS 163
    84                                       <115>
```

Listing 1. »C 64 Centron«. Kernal-Centronics-Schnittstelle ohne Funktionsverlust.
Beachten Sie bitte die Eingabehinweise auf Seite 77.

```
420 POKE 1,53                                <001>
430 PRINT"{CLR,DOWN,RIGHT}BASIC UND C64 CEN
    TRON BEFINDEN SICH":PRINT"JETZT IM RAM.
    "                                        <010>
440 PRINT"{DOWN,RIGHT}DIE FUNKTION KANN GET
    ESTET WERDEN."                           <132>
450 END                                      <067>
1000 I=0                                     <022>
1010 READ A:IF A<0 THEN RETURN               <219>
1020 POKE OF+I,A:I=I+1:GOTO 1010             <105>
1030 END                                     <138>
2000 READ A:POKE OF,A:READ A:POKE OF+1,A:RE
     TURN                                    <234>
3000 READ A:POKE OF,A:READ A:POKE OF+1,A:RE
     AD A:POKE OF+2,A:RETURN                 <135>
31999 DATA 120,160,0,169,224,141,15,64,169,
      96,141,18,64,185,0,255,153,0,255,200   <019>
32000 DATA 208,247,238,15,64,238,18,64       <072>
32001 DATA 173,18,64,201,128,208,234,88,96,
      169,0,133,251,169                      <138>
32002 DATA 96,133,252,169,1,162,8,160,1,32,
      186,255,169,13,162,73,160,64,32,189    <254>
32003 DATA 255,169,251,162,0,160,128,32,216
      ,255,96,67,54,52,32,67,69,78,84,82     <226>
32004 DATA 79,78,47,79,0,-55,169,194,160,25
      4,-55,169,234,160,253,-55,13,163       <119>
32005 DATA 13,83,82,160,160,80,204,82,195,1
      3,204,13,211,160,13,214,13,160,13,43   <001>
32006 DATA 141,-55,160,2,160,6,160,11,160,1
      6,160,13,160,0,160,18,160,7,160,20     <141>
32007 DATA 160,9,169,37,141,4,220,169,64,14
      1,5,220,76,110,255,32,66,89            <113>
32008 DATA 84,69,83,32,82,65,77,13,0,-55,14
      7,13,32,42,32,67,54,52,32,67,69,78     <218>
32009 DATA 84,82,79,78,32,42,32,0,-55,76,95
      ,228,76,212,240,32,0,241,44,12,221     <192>
32010 DATA 48,6,32,151,238,76,68,237,165,14
      9,44,12,221,80,26,72,173,12,220,201    <248>
32011 DATA 219,208,3,108,54,3,104,201,65,14
      4,10,201,192,144,4,41,127,208,2,9,32   <011>
32012 DATA 32,183,228,24,36,163,16,5,169,0,
      141,12,221,88,96,169,13,32,183,228,20
      8                                      <150>
32013 DATA 242,-55,141,1,221,173,13,221,173
      ,0,221,41,251,141,0,221,9,4,141,0,221 <028>
32014 DATA 169,16,44,13,221,240,251,96,-55,
      201,36,240,14,173,12,221,41,127,141    <225>
32015 DATA 12,221,32,151,238,76,39,237,169,
      255,141,3,221,44,12,221,48,13,173,12   <033>
32016 DATA 221,9,128,141,12,221,169,0,32,18
      3,228,24,88,96,120,44,12,221,48,5,88   <045>
32017 DATA 32,54,237,96,72,41,31,201,7,8,17
      3,12,221,40,208,3,9,64,44,41,191,141   <031>
32018 DATA 12,221,104,41,240,201,224,208,21
      5,76,154,228,-55                       <065>
```

Listing 1. »C 64 Centron« (Schluß)

```
10 POKE 56332,219:REM SETZT SCHLUESSEL       <202>
20 POKE 822,60:POKE 823,3:REM SETZT ZEIGER   <207>
30 I=0:RESTORE                               <015>
40 READ A:IF A<0 THEN 100                    <017>
50 POKE 828+I,A:I=I+1                        <036>
60 GOTO 40                                   <041>
100 PRINT"{CLR,7DOWN,7RIGHT}BENUTZERSCHNIT
    TSTELLE AKTIV!!!"                        <077>
110 END                                      <238>
32000 DATA 104,201,255,240,4,72,76,124,228
      ,72,152,72,160,0,185,91,3,32,183,228   <228>
32001 DATA 200,192,37,208,245,104,168,104,
      76,142,228,67,54,52,32,67,69,78,84,8
      2                                      <062>
32002 DATA 79,78,32,77,73,84,32,66,69,78,8
      5,84,90,69,82,83,67,72,78,73,84,84,8
      3                                      <121>
32003 DATA 84,69,76,76,69,-55                <154>
```

Listing 2. Beispiel zur benutzerdefinierten Schnittstelle in »C 64 Centron«

```
1 REM UM IN DEN BEREICH AB $8000            <001>
2 REM ZU VERSCHIEBEN :                      <168>
3 REM                                       <146>
4 REM 10 POKE56,128..                       <177>
5 REM                                       <148>
6 REM 20 OF=24576...                        <091>
7 REM 60 IF PR<>3153...                     <253>
8 REM 95 DATA 0,160,160...                  <095>
9 REM ******************************        <174>
10 POKE 56,96:POKE 55,0:CLR                 <020>
20 OF=32768 : RESTORE                       <043>
30 PRINT"{CLR}READING MC-PROGRAM...":PRINT
   :PR=0                                    <120>
40 FOR N=828 TO 853                         <143>
50 READ D:POKE N,D:PR=PR+D                  <170>
55 NEXT                                     <185>
60 IF PR<>3121 THEN PRINT"FEHLER IN MC-DAT
   AS":STOP                                 <089>
70 PRINT"VERSCHIEBE $E000-$FFFF NACH $6000
   ":PRINT                                  <060>
80 SYS 828                                  <144>
90 DATA 169,0,160,224,133,95,132,96,169,0,
   168,133,90,132,91,169                    <137>
95 DATA 0,160,128,133,88,132,89,76,191,163 <128>
100 BL=1                                    <213>
110 PR=0                                    <242>
120 READ A,E,P :PRINT"BLOCK :"BL;           <178>
130 FOR N=A TO A+E-1                        <244>
140 READ D :POKE N-OF,D :PR=PR+D            <068>
150 NEXT                                    <024>
160 IF PR=P THEN PRINT" O.K.":GOTO 200      <158>
170 PRINT"PRUEFSUMME FALSCH! "PR" STATT "P  <040>
180 PRINT:PRINT"TASTE DRUECKEN":PRINT       <009>
190 POKE 198,0:WAIT 198,1:POKE 198,0        <237>
200 BL=BL+1:IF BL<=4 GOTO 110               <120>
210 PRINT:PRINT"ZEIGER AUF AUSGABE BERICHT
    IGEN"                                   <064>
220 POKE 64822-OF,65:POKE 64823-OF,252 :RE
    M OPEN-VEKTOR                           <048>
230 POKE 64828-OF,167:POKE 64829-OF,252:RE
    M CHKOUT-VEKTOR                         <021>
240 POKE 64834-OF,105:POKE 64835-OF,252:RE
    M CHROUT-VEKTOR                         <024>
900 PRINT:PRINT"AENDERUNGEN DURCHGEFUEHRT.
    ":PRINT:PRINT:END                       <065>
1000 REM BLOCK 1 $E4B8-$E4C8                <114>
1010 DATA 58552,16,2331                     <009>
1020 DATA 32,189,255,165,185,234,234,201,7
     ,240,4,169,2,133,185,96                <047>
1030 REM BLOCK 2 $FC41-$FCCF                <171>
1040 DATA 64577,143,16117                   <148>
1050 DATA 165,186,201,4,240,3,76,74,243,16
     9,1,162,104,160,252                    <116>
1060 DATA 32,184,228,32,74,243,176,8,165,2
     ,41,253,133,2,24                       <236>
1070 DATA 96,165,2,9,2,133,2,24,96,13,72,1
     65,154,201,4                           <045>
1080 DATA 240,3,76,205,241,165,2,41,2,208,
     4,165,154,208,243                      <039>
1090 DATA 173,13,221,41,16,234,234,165,2,4
     1,1,240,23,173,24                      <035>
1100 DATA 208,41,2,234,234,104,201,64,144,
     10,201,192,144,4,41                    <139>
1110 DATA 127,208,2,9,32,72,104,76,172,243
     ,24,96,32,15,243                       <032>
1120 DATA 240,3,76,1,247,32,31,243,165,186
     ,201,4,240,3,76                        <237>
1130 DATA 91,242,165,2,41,2,240,247,165,2,
     41,254,133,2,165                       <038>
1140 DATA 185,41,7,201,7,76,194,254         <217>
1150 REM BLOCK3  $FEC4-$FED3                <040>
1160 DATA 65218,18,1960                     <166>
1170 DATA 208,6,169,1,5,2,133,2,169,255,14
     1,3,221,169,4,76,153,243               <227>
1180 REM BLOCK4 $F399-$F3C0                 <026>
1190 DATA 62361,47,5538                     <199>
1200 DATA 13,0,221,141,0,221,173,2,221,9,4
     ,141,2,221,169                         <239>
1210 DATA 4,76,117,242,141,1,221,173,0,221
     ,41,251,141,0,221                      <143>
1220 DATA 234,234,234,9,4,141,0,221,173,13
     ,221,41,16,240,249,24,96               <008>
1230 REM ENDE                               <125>
```

Listing 3. Centronics-Schnittstelle in Hypra-Perfekt eingebunden

```
10 GOTO 90                                  <252>
20 REM HEX -> DEZ                           <214>
30 SA$=RIGHT$("{4SPACE}"+SA$,4)             <168>
40 SA=0:FOR I=1 TO LEN(SA$):S0=ASC(MID$(SA
   $,I,1)):SA=16*SA+S0-48+(S0>64)*7:NEXT    <195>
50 RETURN                                   <192>
60 REM ADRESSE -> LOW,HIGH                  <123>
70 SH=INT(S4/256):SL=S4-256*SH              <011>
80 RETURN                                   <222>
90 PRINT CHR$(14);"{CLR,2DOWN,7SPACE}C-64
   PARALLEL PROGRAMM"                       <048>
100 PRINT"{7SPACE}----------------------"  <031>
110 PRINT:                                  <065>
120 INPUT "STARTADRESSE: ";SA$:SA=VAL(SA$) <209>
130 IF LEFT$(SA$,1)="$"THEN GOSUB 30        <236>
140 READ EN                                 <166>
150 FOR I=1 TO EN                           <122>
160 READ A,B:IF SA+187>=A AND SA<=B THEN R
    UN                                      <056>
170 NEXT                                    <044>
180 PRINT:PRINT"EINLESEN DES MC-PROGRAMMES
    ":PRINT                                 <032>
190 S1=SA+58:S2=SA+122:S3=SA+37             <253>
200 FOR I = SA TO SA+187                    <090>
210 READ DA                                 <222>
220 POKE I,DA                               <109>
230 NEXT I                                  <177>
240 S4=S2 : GOSUB 70                        <220>
250 POKE SA+2,SH:POKE SA+4,SL               <214>
260 S4=S1 :GOSUB 70                         <240>
270 POKE SA+12,SH:POKE SA+14,SL             <077>
272 S4=S3 :GOSUB 70                         <254>
274 POKE SA+28,SL:POKE SA+29,SH             <094>
280 PRINT:PRINT"WIEDEREINSCHALTEN NACH 'ST
    OP/RESTORE'"                            <184>
290 PRINT:PRINT"MIT{6SPACE}'SYS ";SA;"'"    <088>
300 SYS SA                                  <095>
310 END                                     <183>
320 REM" GESPERRTE BEREICHE                 <090>
330 DATA 4                                  <002>
340 DATA 0,827,1023,2039,40960,49151,53248
    ,65535                                  <134>
350 REM MASCHINENPROGRAMM                   <233>
360 DATA 120,169,192,160,122,141, 33,  3,1
    40, 32,  3,169,192,160, 58,141          <173>
370 DATA 39,  3,140, 38,  3, 88, 32,204,25
    5,160, 20,185, 37,192, 32,210           <248>
380 DATA 255,136, 16,247, 96, 13, 86, 73,
    84, 75, 65, 32, 84, 82, 79, 80          <207>
390 DATA 45, 76, 69, 76, 76, 65, 82, 65, 8
    0, 13, 72,165,154,201,  4,240           <197>
400 DATA  3, 76,205,241,173, 13,221,165,
    2, 41,  1,240, 16,104,201, 64           <253>
410 DATA 144, 10,201,192,144,  4, 41,127,2
    08,  2,  9, 32, 72,104,141,  1          <214>
420 DATA 221,173,  0,221, 41,251,141,  0,2
    21,234,234,234,  9,  4,141,  0          <007>
430 DATA 221,173, 13,221, 41, 16,240,249,
    24, 96, 32, 15,243,240,  3, 76          <095>
440 DATA  1,247, 32, 31,243,165,186,201,
    4,240,  3, 76, 91,242,165,  2           <008>
450 DATA 41,254,133,  2,165,185, 41,  7,20
    1,  7,208,  6,169,  1,  5,  2           <075>
460 DATA 133,  2,169,255,141,  3,221,169,
     4, 13,  0,221,141,  0,221,169          <063>
470 DATA  4, 13,  2,221,141,  2,221,169,
    4, 76,117,242                           <108>
```

Listing 4. Eine frei verschiebbare Centronics-Schnittstelle als Softwarelösung

File-Compactor

Wozu unnötigen Ballast auf Diskette speichern und laden? Der File-Compactor kürzt fast jedes Programm — bis zu 50 Prozent. Einzige Voraussetzung: mindestens eine Basic-Zeile.

Fast alle Programme, egal ob Basic oder Assembler, weisen Folgen von gleichen Zeichen auf. Denken Sie nur an Grafikbilder, die in einem Programm enthalten sein könnten. Diese Zeichenfolgen lassen sich kürzen.

Man geht dazu folgendermaßen vor: Man durchsucht das ganze Programm nach Folgen von gleichen Bytes. Wurde eine Folge gefunden, wird ein Merkzeichen (in diesem Fall $bf oder $cf) an die momentane Stelle geschrieben. Danach der Zeichencode und die Anzahl der Zeichen. Verwechslungen mit eventuell vorkommenden $bf- oder $cf-Codes können dadurch ausgeschlossen werden, indem man vor diese Codes noch mal den Code und für die Anzahl eine »1« schreibt.

Damit die gekürzte Version lauffähig ist, wird vor das eigentliche Programm noch ein »Entpacker« gesetzt, der auch mit abgespeichert werden muß. Von jedem Programm bleibt nach dem Komprimieren nur noch eine Zeile mit einem SYS-Befehl (zum Start des Entpackers) übrig.

Tippen Sie RUN ein, wird vor dem eigentlichen Programmstart das ursprüngliche Programm aus den komprimierten Daten wieder zusammengesetzt; erkenntlich an der dunkelblauen Rahmenfarbe. Das komprimierte Programm wird dazu an das Speicherende verschoben und dann Byte für Byte wieder an die Stelle gesetzt, wo es vor der Behandlung mit dem File-Compactor stand. Stoppen Sie das Basic-Programm nach dem Entpacken mit STOP oder STOP/RESTORE, sehen Sie wieder das normale Basic-Programm in seiner vollen Länge.

Spart Platz auf Disketten

Ein Beispiel: Bestimmte Versionen des Spiels »Apocalypse« sind 153 Blocks lang. Nach der Behandlung mit dem File-Compactor aber nur noch 83 Blocks, also um 46 Prozent kürzer.

Tips zur Benutzung

1. File-Compactor mit dem MSE eingeben und abspeichern.
2. Das so gespeicherte Programm mit LOAD"FILE-COMPACTOR",8 laden und mit RUN starten.
3. Diskette mit dem zu kürzenden Programm einlegen.
4. Namen des Programms eingeben und RETURN drücken. Das Programm wird geladen.
5. Diskette mit genügend freiem Platz einlegen.
6. Namen des gekürzten Programmes eingeben und RETURN drücken. Die gekürzte Version wird gespeichert.

Wollen Sie ein Maschinenprogramm kürzen, das mit LOAD "name",8,1 geladen und mit SYS gestartet wird, müssen Sie das Programm erst mit dem Basic-Start-Generator auf Seite 74 »vorbehandeln«.

(Marcus Breiing/hm)

```
programm : file-compactor  0801 0a9a
--------------------------------------
0801 : 08 08 c1 07 9e 32 30 38  0b
0809 : 38 20 20 46 49 4c 45 20  6f
0811 : 43 4f 4d 50 41 43 a4 52  bf
0819 : 20 4d 2e 42 52 45 49 49  bb
0821 : 4e 47 00 00 00 00 00 a9  66
0829 : 0b 8d 20 d0 8d 21 d0 a9  95
0831 : 36 85 01 a9 51 a2 09 20  2e
0839 : af 08 20 c0 08 a9 01 a2  24
0841 : 08 a0 00 20 ba ff a9 00  f0
0849 : a2 9a a0 0a 20 d5 ff b0  b4
0851 : e2 86 fd 84 fe a9 92 a2  53
0859 : 09 20 af 08 20 c0 08 a9  db
0861 : 01 a2 08 a0 01 20 ba ff  c5
0869 : 20 c0 ff b0 e8 a2 01 20  e7
0871 : c9 ff a9 a7 a2 09 a0 00  8e
0879 : 85 fb 86 fc b1 fb 20 d2  5e
0881 : ff e6 fb d0 02 e6 fc a9  ab
0889 : 9a a2 0a e4 fc d0 02 c5  7d
0891 : fb d0 e9 a9 9a a2 0a 85  96
0899 : fb 86 fc 20 da 08 20 cc  22
08a1 : ff a9 01 20 c3 ff a9 37  0a
08a9 : 85 01 18 4c 34 a8 85 fb  d5
08b1 : 86 fc a0 00 b1 fb f0 06  a8
08b9 : 20 d2 ff c8 d0 f6 60 a2  e7
08c1 : 00 20 cf ff c9 0d f0 09  a0
08c9 : 9d 3c 03 e8 e0 10 d0 f1  18
08d1 : ca 8a a2 3c a0 03 4c bd  df
08d9 : ff 20 35 09 aa 20 46 09  2e
08e1 : 84 02 c0 03 90 2a 98 18  e4
08e9 : 65 fb 85 fb 90 02 e6 fc  db
08f1 : e0 00 f0 10 a9 bf 20 40  a9
08f9 : 09 a5 02 20 40 09 8a 20  10
0901 : 40 09 d0 23 a9 cf 20 40  78
0909 : 09 a5 02 20 40 09 d0 17  27
0911 : e0 bf f0 04 e0 cf d0 0b  73
0919 : 8a 20 40 09 a9 00 20 40  80
0921 : 09 d0 04 8a 20 40 09 a5  58
0929 : fb a6 fc e4 fe d0 02 c5  5d
0931 : fd 90 a6 60 a0 00 b1 fb  f5
0939 : e6 fb d0 02 e6 fc 60 20  a9
0941 : d2 ff a9 ff 60 a0 00 d1  2c
0949 : fb d0 04 c8 d0 f9 88 60  86
0951 : 9b 93 0e 08 12 c6 49 4c  50
0959 : 45 20 c3 4f 4d 50 41 43  6c
0961 : 54 4f 52 20 31 20 20 20  ca
0969 : 20 20 20 31 39 38 35 20  32
0971 : 42 59 20 cd 2e c2 52 45  ee
0979 : 49 49 4e 47 0d 11 11 d0  23
0981 : 52 4f 47 52 41 4d 4d 2d  a5
0989 : ce 41 4d 45 20 20 3e 20  30
0991 : 00 0d 11 11 d3 50 45 49  e6
0999 : 43 48 45 52 4e 20 41 4c  1f
09a1 : 53 20 20 3e 20 00 01 08  ea
09a9 : 1e 08 c1 07 9e 20 28 32  0d
09b1 : 30 38 35 29 20 43 4f 4d  64
09b9 : 50 41 43 54 20 46 43 31  a9
09c1 : ad 4d 54 42 00 00 00 00  72
09c9 : 00 00 00 00 20 53 e4 a2  3f
09d1 : 00 bd 7e 08 9d 3c 03 e8  ea
09d9 : e0 c0 d0 f5 a9 36 85 01  70
09e1 : 8d 20 d0 a9 00 a2 d0 85  4b
09e9 : fb 86 fc a5 2d a6 2e 85  e7
09f1 : fd 86 fe a0 00 f0 04 b1  00
09f9 : fd 91 fb a9 f2 a2 08 e4  21
0a01 : fe d0 02 c5 fd f0 13 a5  a0
0a09 : fd d0 02 c6 fe c6 fd a5  31
0a11 : fb d0 02 c6 fc c6 fb 4c  5c
0a19 : 50 08 a9 01 a2 08 85 2d  d3
0a21 : 86 2e 4c 3c 03 20 9a 03  fb
0a29 : c9 cf d0 11 20 9a 03 c9  a7
0a31 : 00 d0 04 a9 cf d0 1c aa  19
0a39 : a9 00 4c 83 03 c9 bf d0  85
0a41 : 12 20 9a 03 c9 00 d0 04  52
0a49 : a9 bf d0 07 aa 20 9a 03  03
0a51 : 4c 83 03 20 a5 03 a9 00  3d
0a59 : a2 d0 e4 fc d0 02 c5 fb  68
0a61 : d0 c3 a9 37 85 01 20 59  f8
0a69 : a6 4c ae a7 e8 a0 00 91  8d
0a71 : 2d c8 ca d0 fa 88 98 38  96
0a79 : 65 2d 85 2d 90 02 e6 2e  8d
0a81 : 4c 6d 03 a0 00 b1 fb e6  a4
0a89 : fb d0 02 e6 fc 60 a0 00  9f
0a91 : 91 2d e6 2d d0 02 e6 2e  2d
0a99 : 60 ed b1 ca d0 fa a5 62  f6
```

Listing zum File-Compactor. Es muß mit dem MSE eingegeben werden. Beachten Sie bitte dazu die Eingabehinweise auf Seite 77.

Elektronischer Merkzettel

Wenn Sie mit zwei unabhängigen Bildschirmen arbeiten wollen, dann werden Sie von diesem Programm begeistert sein.

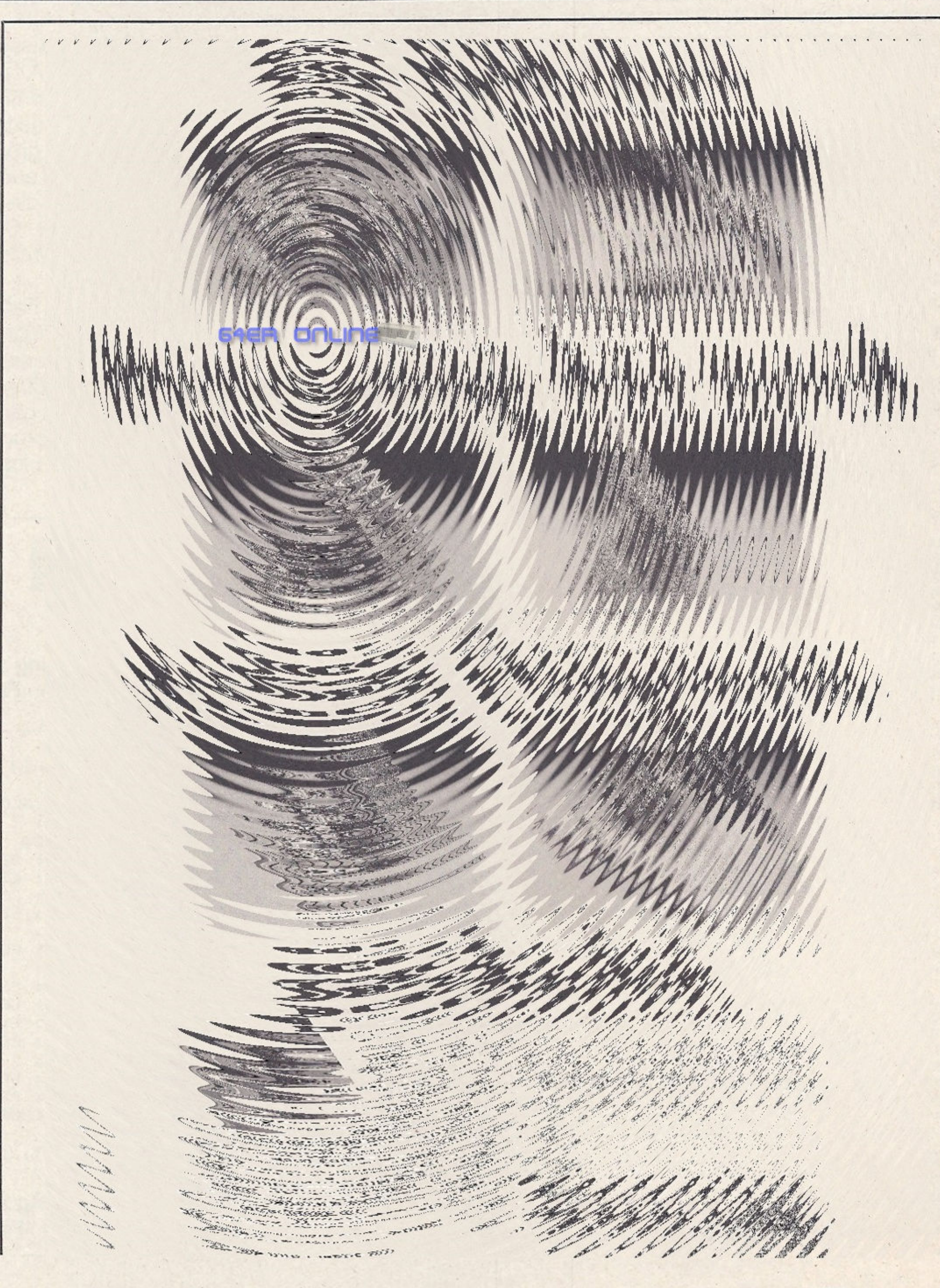
```
10 REM ****************************         <091>
20 REM **        MERK-ZETTEL        **      <127>
21 REM **    ------------------     **      <163>
22 REM ** WRITTEN BY GEORG KRAMER **        <075>
23 REM **      4783 ANROECHTE       **      <189>
24 REM **                          **       <175>
25 REM ****************************         <106>
26 :                                        <084>
30 PRINT CHR$(147):PRINT                    <062>
32 REM **** COPYRIGHT ****                  <152>
35 PRINT SPC(12)"UCCCCCCCCCCCCCI"           <183>
36 PRINT SPC(12)"B MERK-ZETTEL B"           <235>
37 PRINT SPC(12)"JCCCCCCCCCCCCCK"           <176>
38 PRINT:PRINT SPC(8)"WRITTEN BY GEORG KRA
   MER"                                     <219>
39 PRINT:PRINT SPC(13)"(C) 16.2.1985"       <106>
40 PRINT:FOR I=1 TO 40:PRINT"-";:NEXT       <235>
50 PRINT SPC(41)"ZWISCHENSPEICHERN DER BIL
   DSCHIRMSEITE"                            <133>
55 PRINT:PRINT SPC(10)"NEUE TASTENFUNKTION
   :{DOWN,21LEFT}---------------------"     <139>
60 PRINT:PRINT SPC(11)"{RVSON,SPACE}F1{SPA
   CE,RVOFF,SPACE}- BILD WECHSEL"           <139>
80 PRINT:PRINT SPC(54)"<< TASTE >>"         <173>
81 POKE 198,0                               <230>
82 GET A$:IF A$=""THEN 82                   <137>
83 PRINT CHR$(147)SPC(52)"ETWAS GEDULT..."  <237>
84 REM *** DATEN LADEN ***                  <175>
85 FOR I=49152 TO 49415                     <129>
86 READ A:POKE I,A:X=X+A:NEXT               <232>
87 IF X<>38638 THEN PRINT"DATA FEHLER !":E
   ND                                       <217>
88 POKE 49152+214,0:REM + SCHRIFTFARBE +    <035>
89 SYS 49152:REM PRG.-START                 <151>
90 PRINT"{CLR}":PRINT SPC(44)"{RVSON,8SPAC
   E}PROGRAMM AKTIVIERT{5SPACE}"            <066>
91 PRINT SPC(4)"{RVSON,11SPACE}$C000-$C107
   {9SPACE}"                                <112>
92 PRINT SPC(4)"{RVSON,3SPACE}ENDE MIT RUN
   /STOP-RESTORE{3SPACE}"                   <015>
93 END                                      <221>
99 REM **** MASCHINENPROGRAMM ****          <061>
100 DATA 76,251,192,162,0,134,251,162,194,
    134,252,162,232,134,253,162,197         <069>
101 DATA 134,254,162,0,134,182,162,199,134
    ,183,160,0,177,251,145,182,165,252      <219>
102 DATA 197,254,240,22,165,251,201,255,24
    0,24,230,251,165,182,201,255,240        <103>
103 DATA 28,230,182,169,0,201,0,240,224,16
    5,251,197,253,208,228,240,24,169        <114>
104 DATA 255,133,251,230,252,169,0,201,0,2
    40,220,169,255,133,182,230,183,169      <204>
105 DATA 0,201,0,240,216,96,165,203,201,4,
    240,4,76,49,234,96,169,0,141,4,192      <199>
106 DATA 169,4,141,8,192,169,0,141,12,192,
    169,8,141,16,192,169,0,141,20,192       <169>
107 DATA 169,194,141,24,192,32,3,192,169,0
    ,141,4,192,169,199,141,8,192,169        <147>
108 DATA 232,141,12,192,169,202,141,16,192
    ,169,0,141,20,192,169,4,141,24,192      <208>
109 DATA 32,3,192,169,0,141,4,192,169,194,
    141,8,192,169,232,141,12,192,169        <134>
110 DATA 197,141,16,192,169,0,141,20,192,1
    69,199,141,24,192,32,3,192,169,0        <134>
111 DATA 133,251,169,216,133,252,160,0,169
    ,1,145,251,230,251,208,250,166,252      <212>
112 DATA 224,218,240,8,230,252,162,0,224,0
    ,240,232,230,252,160,0,145,251,200      <177>
113 DATA 192,232,208,249,76,49,234,96,234,
    120,169,94,141,20,3,169,192,141         <100>
114 DATA 21,3,88,96                         <238>
```

Listing »Merk-Zettel« für den C 64. Beachten Sie bitte die Eingabehinweise auf Seite 77.

Das Programm »Merkzettel« ist ganz in Maschinensprache geschrieben. Es entstand aus folgendem Problem: Während man ein Programm schreibt, will man Notizen machen, die man häufig auf einem Zettel notiert. Hat man dann erst einmal einige Zettel vollgeschrieben, findet man meistens die erste Notiz nicht gleich wieder. Das kostet Zeit und beansprucht die Nerven. Dieses Programm ermöglicht nun das Führen eines Notizblattes direkt mit dem Computer. Durch Drücken der Funktionstaste F1 wird der Inhalt des Bildschirms mit dem Inhalt des Speichers vertauscht. Sie schreiben also Ihre Notizen auf den Bildschirm, drücken die Taste F1 und schon haben Sie die Information gespeichert. Wenn Sie sie wieder brauchen, drücken Sie einfach erneut die Taste F1, und auf dem Bildschirm erscheinen die vorher abgelegten Informationen, in einer bestimmten Farbe. Diese Farbe können Sie in der Zeile 88 ändern, indem Sie einen anderen Wert als Null in das Register 49366 POKEn.

Das Programm belegt den Bereich von $C000 bis $C0DB. Außerdem werden die nachfolgenden 2 KByte noch zur Zwischenspeicherung benötigt.

(Georg Kramer/ah)



»Fenster«-Befehle für den C 16

Dieses Programm stellt den im C 16-Basic leider nicht vorhandenen »Window«-Befehl zur Verfügung und macht Schluß mit dem umständlichen Hantieren mit den ESC-Funktionen.

Wer im Befehlssatz des Basic 3.5 nach dem Befehl »Window« sucht, wird nicht fündig. Die groß angekündigte moderne Window-Technik läßt doch zu wünschen übrig. Ein Window, also ein Bildschirmfenster, kann nur mit Hilfe der ESC-Taste definiert werden.

So muß man, um ein Window zu setzen, mit dem Cursor an die linke obere Ecke des gewünschten Bildschirmfensters fahren und die Tasten ESC und T drücken. Die rechte untere Ecke des Fensters wird ebenfalls mit dem Cursor gewählt und mit ESC und B gesetzt. Erst jetzt ist das Fenster definiert.

Im Direktmodus ist dieser Aufwand nicht hinderlich, sondern bringt eher Vorteile, da kein Befehl »WINDOW« am Bildschirm erscheint. Im laufenden Programm ist es jedoch sehr umständlich, ein Fenster zu generieren. Es sind sehr viele PRINT-Anweisungen nötig.

Da die Window-Funktion jedoch in vielen Programmen nützlich ist, lohnt es sich schon, diese Funktion komfortabel als Basic-Befehl zur Verfügung zu haben. Ein kleines Maschinenprogramm macht's möglich.

Nach Eintippen des DATA-Laders (Listing 1) sollten Sie das Programm als erstes unbedingt abspeichern, da es sich nach Kontrolle der Prüfsumme selbst initialisiert und anschließend der DATA-Lader gelöscht wird.

Haben Sie das Programm abgespeichert, dann können Sie es mit »RUN« starten und haben kurz danach den neuen Basic-Befehl »FENSTER« zur Verfügung.

Der FENSTER-Befehl hat folgendes Format und kann in jedem Basic-Programm verwendet werden:

FENSTER AZ,AS,EZ,ES

Die Parameter haben folgende Bedeutung:

AZ = **A**nfangs**Z**eile
AS = **A**nfangs**S**palte
EZ = **E**nd**Z**eile
ES = **E**nd**S**palte

Gibt man im Programm beispielsweise die folgende Zeile 10 ein, so wird ein Fenster von Spalte 5 der fünften Zeile bis einschließlich Spalte 35 der 15. Zeile definiert.

10 FENSTER 5,5,35,15

Das Fenster kann durch zweimaliges Drücken der HOME-Taste oder durch Generieren eines neuen Fensters gelöscht werden.

Dem Fenster-Befehl müssen auf jeden Fall vier Parameter folgen. Für die Parameter AZ und EZ sind nur Zahlen zwischen 1 und 25 zulässig. AS und ES dürfen nur zwischen 1 und 40 liegen. Außerdem ist zu beachten, daß AZ auf keinen Fall größer als EZ, und AS nicht größer als ES gewählt wird.

Der Fenster-Befehl darf auch nicht im Direktmodus verwendet werden, aber da ist die normale Vorgehensweise über »ESC« sowieso sinnvoller. Sollten diese Bedingungen nicht beachtet werden, erscheint die Fehlermeldung »SYNTAX ERROR«.

Das Beispielprogramm Fenster-Demo (Listing 2) generiert zufällige Textfenster und füllt diese farbig aus.

Der FENSTER-Befehl verkürzt ein Basic-Programm bei häufiger Anwendung gegenüber der konventionellen Methode über PRINT-Anweisungen ganz erheblich, so daß die vom Maschinenprogramm beanspruchten 255 Byte trotz des mageren C 16-Speichers kaum ins Gewicht fallen dürften.

(Christian Quirin Spitzner/ev)

```
10 REM ***********************************
20 REM *                                 *
30 REM *    FENSTER FUER C 16 & 116      *
40 REM *                                 *
50 REM *    CHRISTIAN QUIRIN SPITZNER    *
60 REM *    GRUBERSTRASSE 53             *
70 REM *    8011 POING                   *
80 REM *                                 *
90 REM ***********************************
100 :
110 :
120 REM *** SPEICHER BEGRENZEN ***
130 :
140 POKE 53,255 : POKE 54,62 : POKE 55,2
55 : POKE 56,62 : CLR
150 :
160 REM *** ERKLAERUNG ***
170 :
180 PRINT"■■"TAB(12)"*** ■FENSTER■ ***"
190 PRINT"■■       FORMAT:
200 PRINT"■       ■FENSTER AZ,AS,EZ,ES■
210 PRINT"■       AZ = ANFANGSZEILE
220 PRINT"       AS = ANFANGSSPALTE
230 PRINT"       EZ = ENDZEILE
240 PRINT"       ES = ENDSPALTE
250 PRINT"■       NICHT IM DIREKTMODUS !
260 :
270 REM DATA'S EINLESEN ***
280 :
290 FOR I=16128 TO 16383
300 :  READ P
310 :  Z=Z+P
320 :  POKE I,P
330 NEXT I
340 IF Z <> 30097 THEN PRINT"■       ■DAT
A ERROR■":END
350 :
360 REM *** INITIALISIEREN ***
370 :
380 SYS 16128
390 :
400 REM *** FENSTER-BEFEHL AUFRUFEN ***
410 :
420 FENSTER 4,1,25,40
430 :
440 REM *** BASIC-LOADER LOESCHEN ***
450 :
460 NEW
470 :
480 REM *** DATA'S ***
490 :
500 DATA 169,024,141,008,003,169,063,141
510 DATA 009,003,162,000,189,205,063,032
520 DATA 210,255,232,224,043,208,245,096
530 DATA 162,000,032,115,004,221,209,063
540 DATA 208,007,232,224,007,208,243,240
550 DATA 006,032,121,004,076,217,139,032
560 DATA 183,251,032,115,004,032,132,157
570 DATA 142,249,063,032,145,148,032,132
580 DATA 157,142,250,063,032,145,148,032
590 DATA 132,157,142,251,063,032,145,148
600 DATA 032,132,157,142,252,063,174,249
610 DATA 063,224,000,208,003,076,161,148
620 DATA 224,024,016,249,174,251,063,224
630 DATA 026,016,242,236,249,063,048,237
640 DATA 174,250,063,224,000,240,230,224
650 DATA 041,016,226,172,252,063,224,041
660 DATA 016,219,236,250,063,048,214,162
670 DATA 000,222,249,063,232,224,004,208
680 DATA 248,173,249,063,133,205,173,250
690 DATA 063,133,202,169,027,032,210,255
700 DATA 169,084,032,210,255,173,251,063
710 DATA 133,205,173,252,063,133,202,169
720 DATA 027,032,210,255,169,066,032,210
730 DATA 255,169,019,032,210,255,032,193
740 DATA 251,032,121,004,076,217,139,000
750 DATA 000,000,000,000,000,147,013,032
760 DATA 130,070,069,078,083,084,069,082
770 DATA 132,032,040,067,041,032,067,072
780 DATA 082,073,083,084,073,065,078,032
790 DATA 081,085,073,082,073,078,032,083
800 DATA 080,073,084,090,078,069,082,013
810 DATA 013,000,000,000,000,000,000,000

READY.
```



Listing 1. Der DATA-Lader zum »Fenster«-Befehl

```
100 REM FENSTER-DEMO
110 :
120 COLOR 0,1 : COLOR 4,1 : SCNCLR
130 :
140 DO
150 : AZ = INT(RND(TI)*12)+1
160 : EZ = AZ*2+1
170 : AS = INT(RND(TI)*20)+1
180 : ES = AS*2
190 :
200 : FENSTER AZ,AS,EZ,ES
210 :
220 : PRINT"■";
230 : COLOR 1,RND(TI)*15+2,RND(TI)*8
240 : E = (ES-AS+1)*(EZ-AZ+1)-1
250 : FOR I=1 TO E
260 :   PRINT"■ ■";
270 : NEXT
280 : PRINT"■■";CHR$(148);" ■";
290 LOOP

READY.
```

Listing 2. Ein Demo-Programm zum »Fenster«-Befehl

VC 20-Steckmodule auf Diskette

Wer ein Programmierhilfe-Modul wie zum Beispiel den »Super-Expander« und gleichzeitig eine 64-KByte-Speichererweiterung besitzt, der steht vor dem Problem, daß immer nur eines von beiden Modulen eingesetzt werden kann. Man hat also die Wahl zwischen erweitertem Speicher und komfortabler Programmierung.

Doch das muß nicht so sein. Wenn Sie ein Diskettenlaufwerk zur Verfügung haben, dann können Sie das Modul einfach auf Diskette abspeichern und bei Bedarf in den Steckmodulbereich ($A000 bis $BFFF) der 64-KByte-Erweiterung laden. Gehen Sie dabei wie folgt vor:

Bei eingesteckter Supererweiterung wird zunächst der Basic-Anfang durch

POKE 43,0:POKE 44,160

auf $A000, also auf den Steckmodulanfang, gesetzt. Nun wird noch der Zeiger auf das Basic-Ende verbogen:

POKE 45,0:POKE 46,192

Das Modul kann jetzt mit »SAVE "»Name",8« auf Diskette gespeichert werden. Bei Bedarf kann es jederzeit wieder mit »LOAD "Name",8,1« geladen werden, vorausgesetzt natürlich, man hat RAM im Bereich $A000 bis $BFFF zur Verfügung. Die Supererweiterung wird mit »SYS 64802« eingeschaltet und meldet sich dann mit 28023 freien Speicherbytes. Ausgeschaltet werden kann sie mit »SYS 64818«.

Natürlich ist das Laden von Diskette umständlicher, als gleich mit dem Steckmodul zu arbeiten. Der große Vorteil liegt aber darin, daß man jetzt über die Befehle der Supererweiterung und gleichzeitig den vollen Speicherausbau verfügt.

Es muß noch darauf hingewiesen werden, daß sich Spielemodule mit dieser Methode in der Regel nicht kopieren lassen, da sie erstens mit einem Autostart versehen sind und zweitens zusätzliche Schutzmaßnahmen eingebaut haben. Die hier beschriebene Methode funktioniert nur zusammen mit einer Diskettenstation, da das Betriebssystem des VC 20 sich weigert, den Steckmodulbereich auf Kassette abzuspeichern.

(Günter Büntemeyer)

Es muß nicht immer »READY.« sein ...

Wenn Sie sich darüber ärgern, daß der C 64 nach jedem ausgeführten Befehl sein stupides »READY.« auf den Bildschirm schreibt, dann geben Sie doch die beiden folgenden Zeilen im Direktmodus ein:

FOR I=40960 TO 49151 : POKE I,PEEK(I) : NEXT : POKE 1,54 : REM Basic-Interpreter ins RAM laden

FOR I=41848 TO 41853 : POKE I,32 : NEXT : REM READY-Meldung überschreiben

Wenn Sie jetzt irgendeinen Befehl eingeben, erscheint kein »READY.« mehr. Natürlich können Sie aber auch einfach den Text ändern:

A$="HALLO." : FOR I=1 TO 6 : POKE 41847+I,ASC(MID$(A$,I,1)) : NEXT

Nach Eingabe dieser Zeile meldet sich der Interpreter nach jeder Eingabe mit »HALLO.«. Sie können jeden beliebigen Text wählen, vorausgesetzt, er ist maximal sechs Zeichen lang.

(Andreas Scharrer)

Simons Basic und Turbo-Tape

Das langsame Laden von der Datasette ist ein gravierender Nachteil von Simons Basic, der sich aber mit »Turbo Tape« beheben läßt. Nun überschneiden sich zwar Simons Basic und Turbo-Tape in bestimmten Speicherbereichen und bei der Zeropage-Belegung, aber das läßt sich mit wenig Aufwand beheben. Wer sich an die folgenden Anweisungen hält, wird künftig auch bei Simons Basic im Turbo-Tape-Modus laden und abspeichern können. Die Anweisungen gelten für alle Turbo-Tape-Versionen, die einen Basic-Kopf haben und über »SYS 50000« reaktiviert werden können. Die Anfangsadresse des ungekürzten Turbo-Tape ist A=3032. Bei anderen Anfangsadressen sollte der entsprechende Wert im Umschreibeprogramm verwendet werden.

(1) Umschreiben von Turbo-Tape

Durch folgende Manipulationen wird Turbo-Tape in einen anderen Speicherbereich verlegt:

— Simons Basic laden und starten.

— Turbo-Tape laden ohne zu starten.

— Eingabe der folgenden Zeilen im Direktmodus:

A=3032 : B=PEEK(45) + 256 * PEEK(46) : D=0 : DIM A(100)

FOR X=A TO B : C=PEEK(X) : IF C>19 AND C<198 THEN D=D+1 : A(D)=X : NEXT : ELSE : NEXT

H=141 : G = 169

FOR X=1 TO D : C=A(X) : F=PEEK(C-2) : IF F=32 OR F=H OR PEEK(C-1)=G THEN POKE C,PEEK(C)-96: NEXT : ELSE : NEXT

— Turbo-Tape ist jetzt fertig umgeschrieben und sollte zunächst abgespeichert werden.

(2) Der Umgang mit Simons Turbo-Tape

— Erst Simons Basic, dann das umgeschriebene Turbo-Tape laden und starten.

— Um mit Simons Basic normal arbeiten zu können, muß es mit »SYS 32778« reaktiviert werden. Ein eventuell vorhandenes Programm erhält man mit dem Befehl »OLD« zurück.

— Zum Laden oder Abspeichern von Programmen wird Turbo-Tape mit »SYS 25424« aktiviert.

— Simons Basic und Turbo-Tape werden also je nach Bedarf wechselseitig eingeschaltet.

— Der Basic-Speicher sollte auf $6000 eingegrenzt werden, damit Turbo-Tape nicht überschrieben wird.

Wenn man sich genau an diese Hinweise hält, dann steht einem Arbeiten mit Turbo-Tape auch bei Simons Basic nichts mehr im Wege. Es gibt mittlerweile auch neue Versionen von Turbo-Tape und Simons Basic, die problemlos von Anfang an zusammenarbeiten.

(Klaus Holthausen)

Tips für VC 20 und Datasette

POKE 37148,252 Datasetten-Motor einschalten
POKE 37148,254 Datasetten-Motor ausschalten
WAIT37148,2,2 Wartet auf Drücken einer Datasetten-Taste
WAIT 37151,64 Wartet, bis »Play«-Taste ausrastet

(Thomas Hahn)

Textomat mit Centronics-Drucker

Mit wenig Aufwand und zudem sehr preiswert ist es möglich, das Textverarbeitungsprogramm »Textomat« von Data-Becker ohne teures Hardware-Interface an einen Centronics-Drucker (zum Beispiel Epson RX) anzuschließen.

Als Software-Schnittstelle wird dabei das Treiberprogramm von Arnd Wängler und Thomas Krätzig aus der Ausgabe 6/84 des 64'er-Magazins verwendet und das dazugehörige, im gleichen Artikel beschriebene User-Port-Kabel, das ein paar Mark in Anspruch nimmt.

Die Aufgabe besteht nun darin, Textomat irgendwie an diese Schnittstelle anzupassen. Zur Datenübertragung mit dem Wängler/Krätzig-Interface muß ein Druckerfile zur Geräteadresse 4 geöffnet werden, das keine Sekundäradresse oder

allenfalls die Sekundäradresse 128 enthalten darf. Da Textomat aber nur Sekundäradressen zwischen 1 und 14 zuläßt, muß die Anpassung extern erfolgen. Dabei behilft man sich mit einem kleinen Trick:

Mit dem Druckermenü des Textomat wird ein Druckerprogramm zu einer anderen Sekundäradresse erstellt und auf Diskette abgelegt.

Dieses »Programm« wird nach Aus- und Wiedereinschalten des C 64 wie ein normales Basic-Programm mit »LOAD "Drucker",8« geladen. Der LIST-Befehl darf aber auf keinen Fall verwendet werden, da es sich um kein eigentliches Programm handelt. Statt dessen gibt man nach dem Laden »POKE 2048+255,128« ein, wodurch die Sekundäradresse auf 128 geändert wird. Anschließend wird das so geänderte Drucker-File mit »SAVE ":Drucker",8« wieder auf die Diskette zurückgeschrieben. Die für den Epson-Drucker erforderliche Anpassung der Commodore-Zeichen an den Standard-ASCII-Zeichensatz kann über die Druckeranpassung des Textomat erfolgen.

Um nun mit Textomat und dem Centronics-Treiber zu arbeiten, muß stets als erstes das Treiberprogramm aus der Ausgabe 6/84 geladen und mit »SYS 724« initialisiert werden. Dann lädt man das Textomat-Hauptprogramm mit »LOAD "T.OBJ",8,1« und startet es mit »SYS 4108«. An diesen SYS-Befehl sollte man sich beizeiten gewöhnen, da der normale Start mit SYS 4096 den Kopierschutz aktiviert, der das Directory jeder »unbefugten« Diskette löscht.

Der Centronics-Drucker kann jetzt wie ein Commodore-Drucker angesprochen werden, nachdem die ASCII-Anpassung und die Anpassung der Steuerzeichen vorgenommen wurde.

Zu bemerken wäre noch, daß diese Vorgehensweise mit der 83'er Textomat-Version erprobt wurde und das Funktionieren mit neueren Versionen nicht garantiert ist. (Werner Weigelt)

POKEs für den VC 20

POKE 199,1 Wirkung wie RVS ON
POKE 199,0 Wirkung wie RVS OFF
POKE 214,X Setzt Cursor um X Stellen nach unten
POKE 774,0 Listschutz (Listet nur Zeilennummern)
POKE 774,27 Listschutz (Listet nur Unsinn)
POKE 774,26 Listschutz aus
POKE 775,1 Listschutz (VC 20 stürzt ab)
POKE 775,200 Listschutz (»Error« nach erster Zeile)
POKE 775,199 LIST wieder normal
POKE 776,221 Verweigert Befehlsausführung
POKE 777,1 Cursor »HOME« nach jedem Befehl
POKE 777,199 Hebt vorherigen POKE wieder auf
POKE 788,194 RUN/STOP-Taste abschalten
POKE 788,191 RUN/STOP-Taste wieder in Betrieb
POKE 792,2 RESTORE-Taste abschalten
POKE 792,173 RESTORE-Taste wieder in Betrieb
POKE 808,128 RUN/STOP und RESTORE abschalten
POKE 808,112 RUN/STOP und RESTORE wieder in Betrieb
POKE 36864,X Horizontale Bildschirmzentrierung (X=0-59)
POKE 36865,X Vertikale Bildschirmzentrierung (X=0-156)
POKE 36866,X Anzahl Bildschirmspalten (X=128 + Spalten)
POKE 36867,X Anzahl Bildschirmzeilen (X=2-46)
POKE 36869,242 Umschalten auf Kleinschriftmodus
POKE 36869,240 Umschalten auf Großschrift/Grafik
POKE 37148,251 Datasettenmotor aus
POKE 37148,252 Datasettenmotor ein
POKE 37150,2 Setzt RESTORE-Taste außer Betrieb
POKE 37150,130 Setzt RESTORE-Taste wieder in Betrieb

(Herbert Lickes)

Reset-Helfer für C 64

Das Betriebssystem des C 64 enthält ab der Adresse $FD02 ein Unterprogramm, das im Steckmodulbereich ab $8000 nach der Zeichenfolge »CBM80« sucht. Nach dem Einschalten des Computers oder nach einem Reset wird dieses Unterprogramm jedesmal aufgerufen. Werden ab der Adresse $8003 die Zeichen »CBM80« gefunden, dann wird nicht zum Basic-Start gesprungen, sondern das Betriebssystem nimmt an, daß ein Modul eingesteckt ist, holt sich aus der Speicherzelle $8000/$8001 die Startadresse des Modulprogramms und verzweigt dorthin.

Das kleine »Reset-Helfer«-Programm (Listing) nutzt dies aus, indem es die genannten Speicherstellen in geeigneter Weise abändert. Es schreibt die »CBM80«-Zeichenfolge ab $8003 ins RAM und läßt die Speicherstellen $8000/$8001 auf den Basic-Warmstart zeigen.

Wenn man jetzt einen Reset auslöst, kommt man wie gewohnt aus allen »abgestürzten« Programmen heraus, ein vorhandenes Basic-Programm bleibt aber erhalten.

(Henning Zipf)

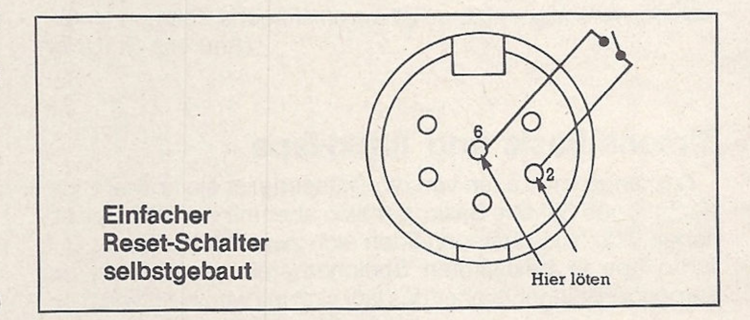
```
100 REM *** RESET-HELFER C 64 ***            <002>
110 REM                                      <253>
120 REM HENNING ZIPF                         <071>
130 REM KIRCHSTR. 8                          <225>
140 REM 6086 RIEDSTADT 5                     <200>
150 REM TEL. (06158)/72453                   <193>
160 REM                                      <047>
170 FOR I=1 TO 9                             <052>
180 READ D                                   <127>
190 POKE 32767+1,D                           <169>
200 NEXT I                                   <147>
210 POKE 53280,14:POKE 53281,6               <050>
220 PRINT CHR$(147);CHR$(5)                  <177>
230 PRINT" C 64 CHANGED RESET VECTOR"        <176>
240 PRINT                                    <137>
250 PRINT" 64 K RAM SYSTEM 38911 BASIC BYT
    ES FREE"                                 <098>
260 PRINT                                    <158>
270 NEW                                      <177>
280 DATA 0,0,255,0,195,194,205,56,48         <211>
290 REM ERST SAVE, DANN RUN !                <130>
```

Listing »Reset-Helfer« für den C 64

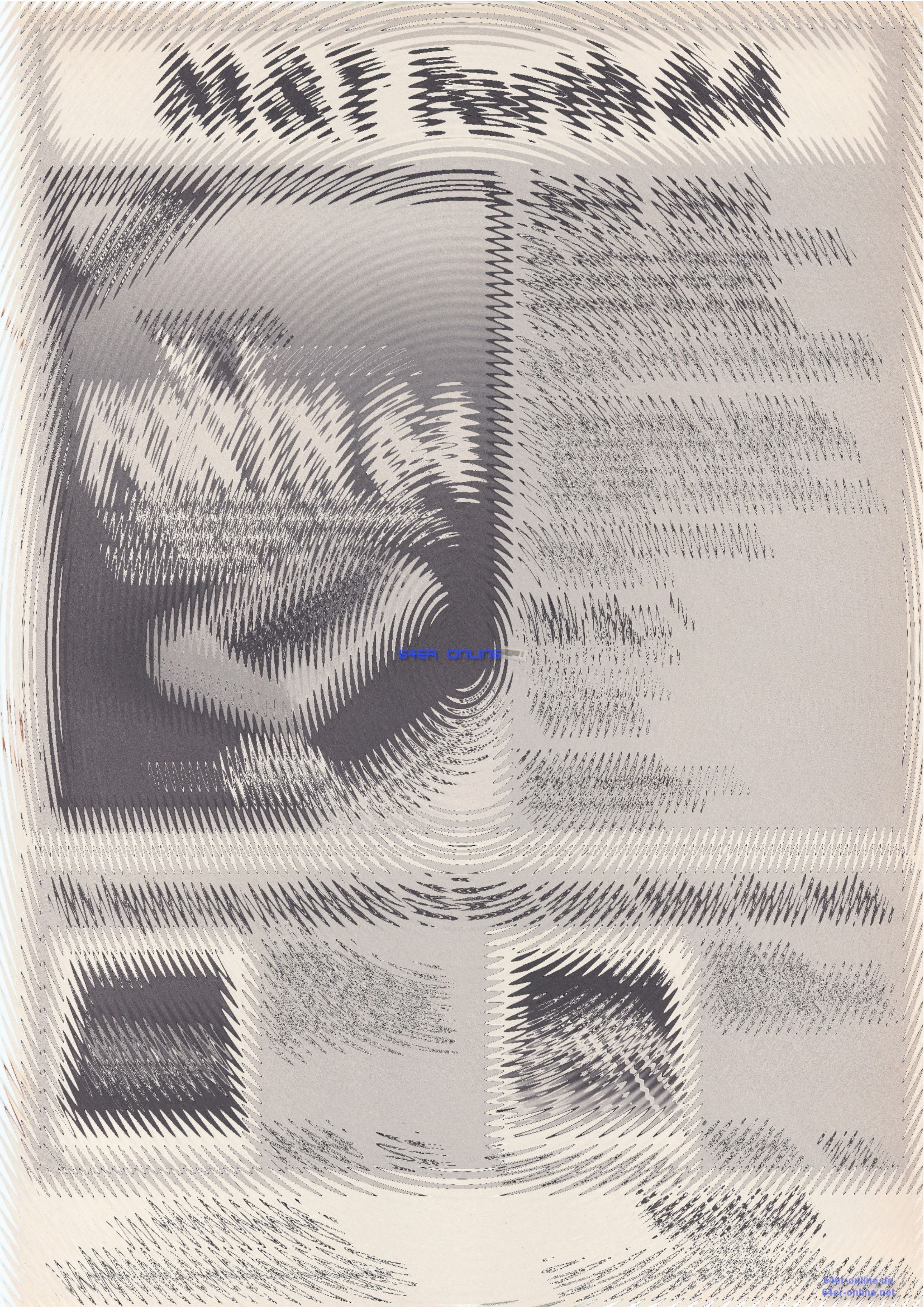
Einfacher Reset-Schalter selbstgebaut

Sehr preiswert und einfach kommt man im Selbstbau zu einem Reset-Schalter, wenn man sich im Elektronikfachgeschäft einen sechspoligen Diodenstecker und einen Miniatur-Tastschalter besorgt. Der Tastschalter wird zwischen die Pins 2 und 6 des Diodensteckers gelötet (siehe Bild). Zur Pinbelegung vergleiche auch Seite 142 im C 64-Handbuch. Der so präparierte Diodenstecker wird nun in den seriellen Port des C 64 beziehungsweise VC 20 gesteckt. Falls der Port bereits von Floppy oder Drucker belegt ist, kann der Stecker natürlich auch in die freie Buchse des entsprechenden Zusatzgerätes eingesteckt werden. Auf Knopfdruck wird nun in jedem Fall wieder der Einschaltzustand des Computers hergestellt.

Die Kosten für Stecker und Taste betragen zusammen etwa 4 Mark. (Henning Zipf)



Einfacher Reset-Schalter selbstgebaut



Die Floppy 1541

Schon seit längerem angekündigt, ist es dieser Tage endlich auf den Markt gekommen: Das Buch über die 1541-Floppy von Karsten Schramm, der den Lesern des 64'er-Magazins als Autor des Floppy-Kurses kein Unbekannter ist. Hier plaudert ein Profi aus seiner Trickkiste. Denn bei diesem Floppy-Buch wurde der Schwerpunkt auf die Themen gelegt, bei denen andere aufhören.

Und so erfährt man dann in lockerem Stil, was in dem Diskettenlaufwerk wirklich abläuft, und wie man es geschickt manipulieren kann. Kein Blatt vor den Mund genommen wird, wenn es um Software-Schutzmethoden geht: Vielleicht ein Ärgernis für den einen oder anderen Softwarehersteller, aber nützlich für den ambitionierten Anwender, der seine selbstgeschriebene Software kopiersicher machen will.

Dennoch ist das vorliegende Buch nicht nur für Profis geeignet: Wer bisher nur die Befehle LOAD und SAVE mit seiner 1541 in Verbindung bringen konnte, der erfährt hier, wie man sequentielle, relative und Direktzugriffs-Dateien realisieren und ausnutzen kann.

Einige der weiteren Themen: Fehler im Commodore-DOS werden schonungslos offengelegt. Der serielle Bus wird unter die Lupe genommen und nach Hypra-Load-Manier beschleunigt. Methoden zur Rettung von verlorengegangenen Daten und fehlerhaften Blöcken werden vorgestellt. Und dies sind noch längst nicht alle der angesprochenen Bereiche.

Das Allerbeste an diesem Buch ist allerdings das dokumentierte Listing des 1541-ROMs. Es ist fast zu schön um wahr zu sein, wie gründlich die Dokumentation vorgenommen wurde, denn sie läßt sich fast schon wie ein zusammenhängender Text lesen. Praktisch jeder einzelne Maschinenbefehl wurde mit einem erläuternden Text versehen, weiter gibt es zu jeder der rund 400 Einzelroutinen des DOS einen kleinen einleitenden Text, dem dann die ausführliche Dokumentation neben dem Assembler-Listing folgt.

Die Dokumentation, die fast die Hälfte des Buches in Anspruch nimmt, wird von einer ebenso ausführlichen RAM-Belegung ergänzt. Mehrere nützliche Programme und ein ausführliches Stichwortverzeichnis runden das äußerst positive Gesamtbild ab. In Vorbereitung ist eine Diskette mit allen abgedruckten und einigen zusätzlichen Programmen. Das eindeutige Urteil: Ein Floppy-Buch, wie man es sich besser kaum vorstellen kann. Von seiner Qualität her gesehen hat es gute Chancen, **das** Standardwerk über die 1541 zu werden.

(Boris Schneider/ev)

Info: Karsten Schramm, Die Floppy 1541, Markt & Technik 1985, 430 Seiten, ISBN 3-89090-098-4, 49 Mark

Grafik & Musik auf dem Commodore 64

Das Thema Nummer Eins auf dem Commodore 64, zumindest was die Bücherproduktion angeht, ist immer noch die Grafik. Im vorliegenden Buch »Grafik & Musik auf dem Commodore 64« wird dieses Thema zusammen mit den akustischen Fähigkeiten sehr ausführlich und mit vielen Beispielen versehen erklärt. Dabei werden nur Basic-Grundkenntnisse vorausgesetzt. Gerade den Anfängern, die vom original Commodore-Handbuch stark enttäuscht worden sind, wird hier auf fast 340 Seiten einiges geboten. Von der einfachen PRINT-Grafik über Sprites und hochauflösende Grafik im Multicolormodus, von einfachen Piepston über Klangeffekte bis zum Musikstück und der Kombination dieser Dinge, wird die Thematik hier voll ausgeschöpft.

Sehr gefallen hat dabei auch der lockere Stil, in dem das Buch gehalten ist. Außerdem wird bewiesen, daß man nicht immer erst graue Theorie von der Praxis lernen muß. Ein Beispiel: Um die Programmierung von Sprites zu verstehen, muß man sich nicht erst durch langwierige Erklärungen des Binärsystems kämpfen, sondern arbeitet vorerst mit einem Spriteentwurfsblatt, das alle wichtigen Daten enthält. Natürlich wird später genau auf die Arbeitsweise des VICs eingegangen, das Binärsystem bleibt ebenfalls nicht unbeachtet. Es kann dann jedoch anhand des schon ausgeführten Beispiels gelernt werden.

Einen einzigen kleinen Haken hat das Ganze aber doch: Da hier fast ausschließlich auf Basic-Ebene gearbeitet wird, arten manche Programme geradezu in POKE- und DATA-Orgien aus, die man aber dank der guten Dokumentation in den Griff kriegen kann. Viele Tabellen, Entwurfsblätter und Registerübersichten bilden einen sehr umfangreichen Anhang, den selbst der inzwischen weit fortgeschrittene Programmierer gerne als Nachschlagewerk benutzen wird. Alles in allem ein empfehlenswertes Buch, für das die 38 Mark sicherlich nicht zu viel sind.

(Boris Schneider)

Info: Stan Krute, Grafik & Musik auf dem Commodore 64, Markt & Technik 1984, zirka 340 Seiten, 38 Mark

Spaß an Mathe mit dem Commodore 64

Endlich einmal ein Buch, das die Mathematik für Computer nicht von der bierernsten Seite her angeht. Das Konzept von »Spaß an Mathe mit dem Commodore 64« hat denn auch die Mathematik nur als Aufhänger: Tatsächlich wird gezeigt, wozu man den Computer außer »Space Invaders« und »Impossible Mission« noch benutzen kann.

Daß dabei dann auch noch eine Vielzahl gut dokumentierter und wirklich nützlicher Programmlistings herausspringt, ist ein weiterer Pluspunkt. Die wichtigsten Listings sind eine Grafik- und eine Mathematik-spracherweiterung, die sich durchaus mit kommerziell vertriebener Software messen können.

Einziger Nachteil: Die beiden arbeiten nicht zusammen. Dafür ist das Eintippen der doch recht umfangreichen Listings recht komfortabel gehalten worden: Im Anhang des Buches befindet sich ein Programm, das die Eingabe enorm erleichtert und gegen Fehler absichert.

Auf den beiden Spracherweiterungen aufbauend werden dann einige mathematische Probleme behandelt, die nicht gerade zum Schulalltag gehören, so zum Beispiel magische Zahlenquadrate, Warteschlangen- oder Populationsprobleme. Auch ein Funktionsplotter darf natürlich nicht fehlen. Und damit man auch wirklich was lernt, sind alle Listings vollständig dokumentiert (auch die Assemblerlistings).

Zusätzlich werden stapelweise Grundlagen über den C 64 und zahlreiche Tips und Tricks vermittelt. Beim Funktionenplotter erfährt man zum Beispiel, wie man Fehlermeldungen abfangen kann, oder wie sich selbstverändernde Programme per Tastaturpuffer programmieren lassen.

Man tut sich also nicht schwer, später eigene, leistungsfähige mathematische Programme zu schreiben. Natürlich darf man nicht als totaler Anfänger an das Buch herangehen, man sollte sich schon etwas mit den Basic-Befehlen auskennen, und auch den Mut haben, erste Kontakte mit Maschinensprache aufzunehmen.

Dann bringt einem dieses Buch aber eine ganze Menge, lernt man doch seinen Commodore 64 am praktischen Beispiel kennen. Man merkt deutlich, daß hier als Autor ein Lehrer am Werke war.

(Boris Schneider)

Info: Holger Danielsson, Spaß an Mathe mit dem Commodore 64, Sybex 1985, 280 Seiten, Preis: 32 Mark

Mathe für die Oberstufe. ...aufgezeigt am Beispiel Commodore 64

So lautet der recht vielversprechende Titel eines neuen Buches aus dem IWT-Verlag, das auch durch einen dem Schülergeldbeutel angepaßten Preis (28 Mark) besticht.

Aber damit sind leider auch fast alle positiven Seiten dieses Buches aufgezählt. Man erhält nämlich nichts weiter als 14 undokumentierte und teilweise recht lange Programmlistings minderer Qualität.

Ein Beispiel: »Kurvendiskussion für Polynome zweiten Grades«. Wer die 13 Druckseiten abgetippt hat, kann sich nun zu jedem solchen Polynom die Symmetrieeigenschaften, Nullstellen, Extremwerte und Wendepunkte sowie das Monotonie- und Krümmungsverhalten ausgeben lassen.

Dummerweise hat ein Polynom zweiten Grades jedoch weder Wendepunkte noch ist es punktsymmetrisch, so daß die entsprechenden Erklärungen und Berechnungen im Programm völlig nutzlos sind.

Ein Nutzen daraus könnte allenfalls vielleicht gezogen werden, wenn man das Programm selber auf andere Funktionen erweitert oder umschreibt, was allerdings durch das Fehlen jeglicher Dokumentation stark erschwert wird.

Es geht aber noch weiter: Man scheute sich nicht, als weiteres Listing eine »Kurvendiskussion mit Integration für Polynome zweiten Grades« abzudrucken. Man ahnt was kommt: Dasselbe Programm noch einmal, dann allerdings um einen Unterprogrammpunkt und auf 20 Druckseiten erweitert.

Vorhanden ist übrigens noch eine Kurvendiskussion von Polynomen dritten Grades (ohne Integration). Eine anspruchsvolle Kurvendiskussion kann mit den abgedruckten Programmen nicht durchgeführt werden. Ähnlich verhält es sich auch mit den diversen Ableitungs- und Integrationsprogrammen, die jeweils nur recht uninteressante und zudem leichte Einzelfälle behandeln. Und auch die Programme, die sich mit der Matrizenrechnung beschäftigen, decken nur einen kleinen Bereich ab. Und schließlich sei zu den begleitenden Texten gesagt, daß sie viel zu kurz und oberflächlich sind.

Langer Rede kurzer Sinn: Die 28 Mark wären bei einer Mathematiknachhilfe besser angelegt.

(Boris Schneider)

Info: J. Merget, Mathe für die Oberstufe, IWT-Verlag 1984, 150 Seiten, 28 Mark

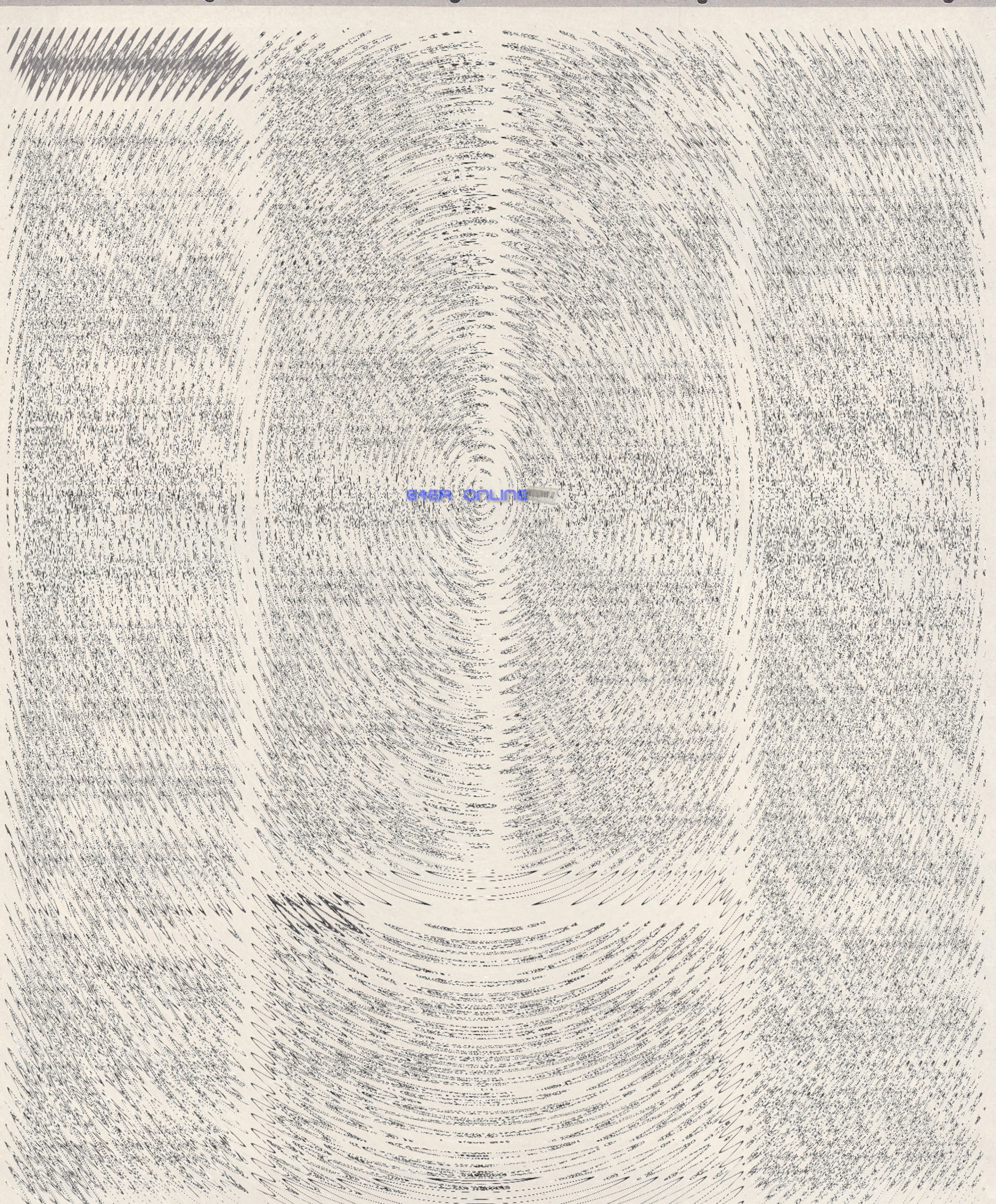


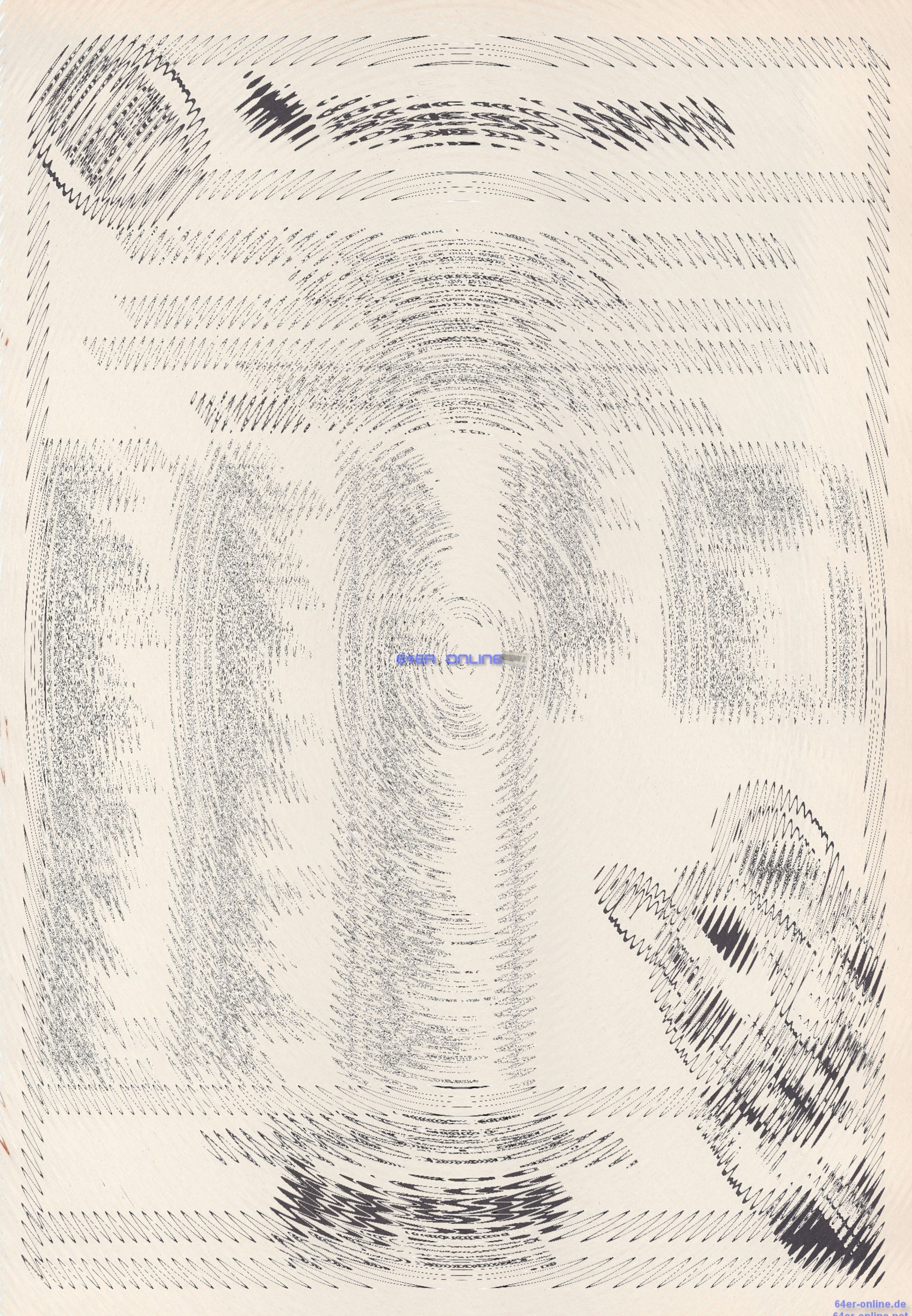
COMPUTER-MARKT

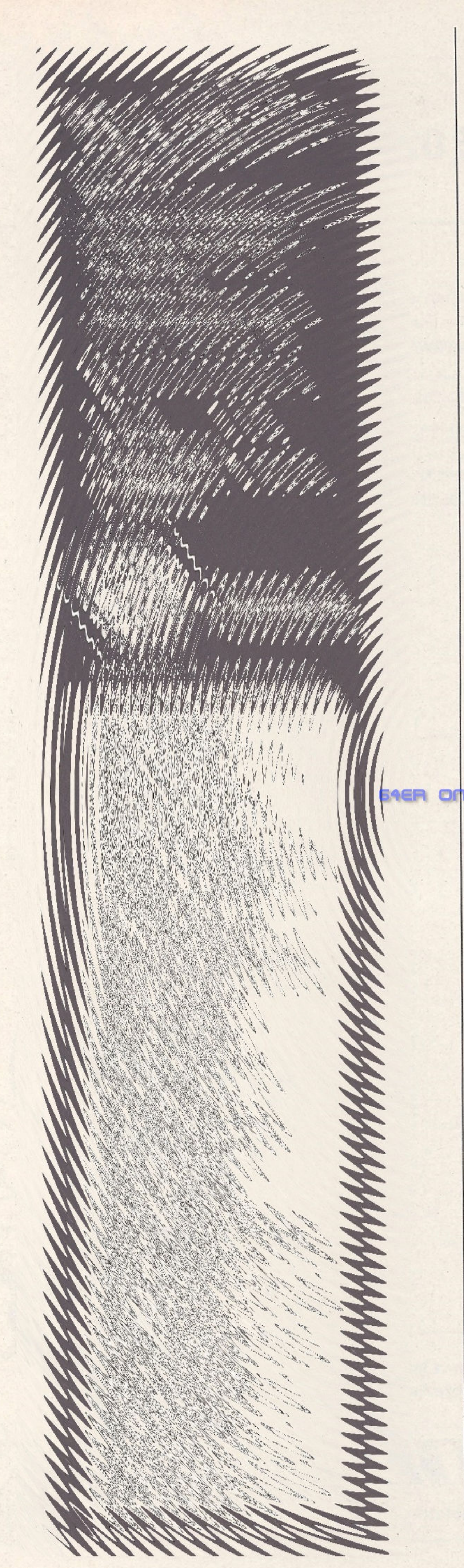
Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64er« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **August-Ausgabe** (erscheint am 19. Juli 85): Schicken Sie Ihren Anzeigentext bis zum 24. Juni 85 (Eingangsdatum beim Verlag) an »64er«. Später eingehende Aufträge werden in der **September-Ausgabe** (erscheint am 16. August 85) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64er« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 11,— je Zeile Text veröffentlicht.

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

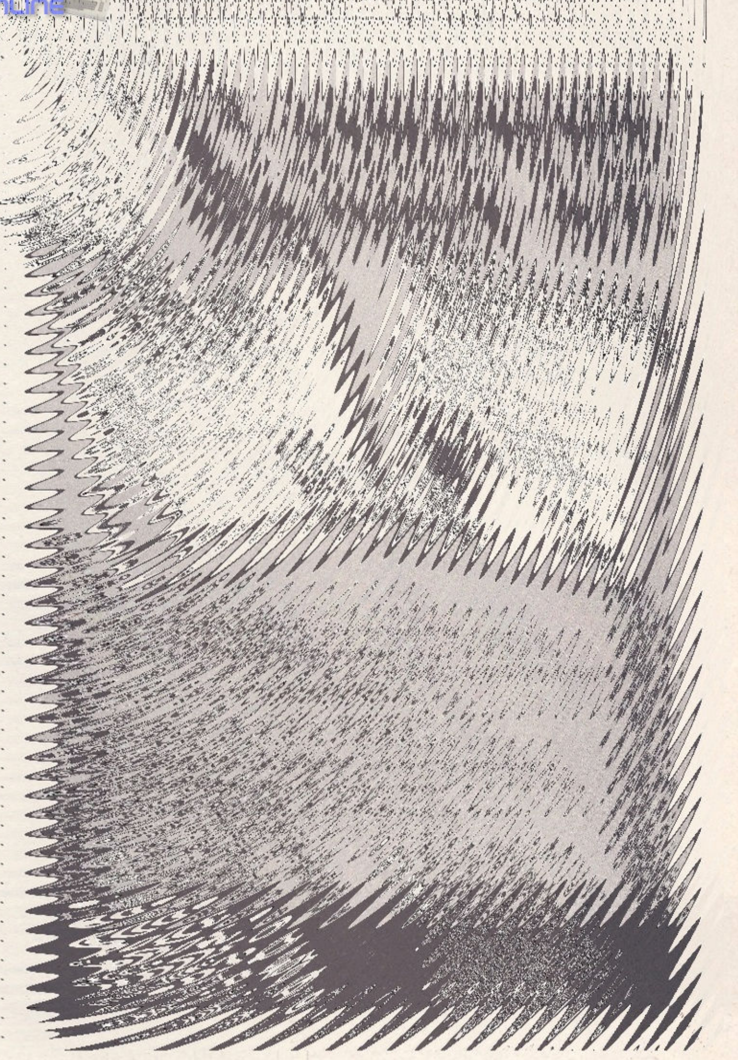




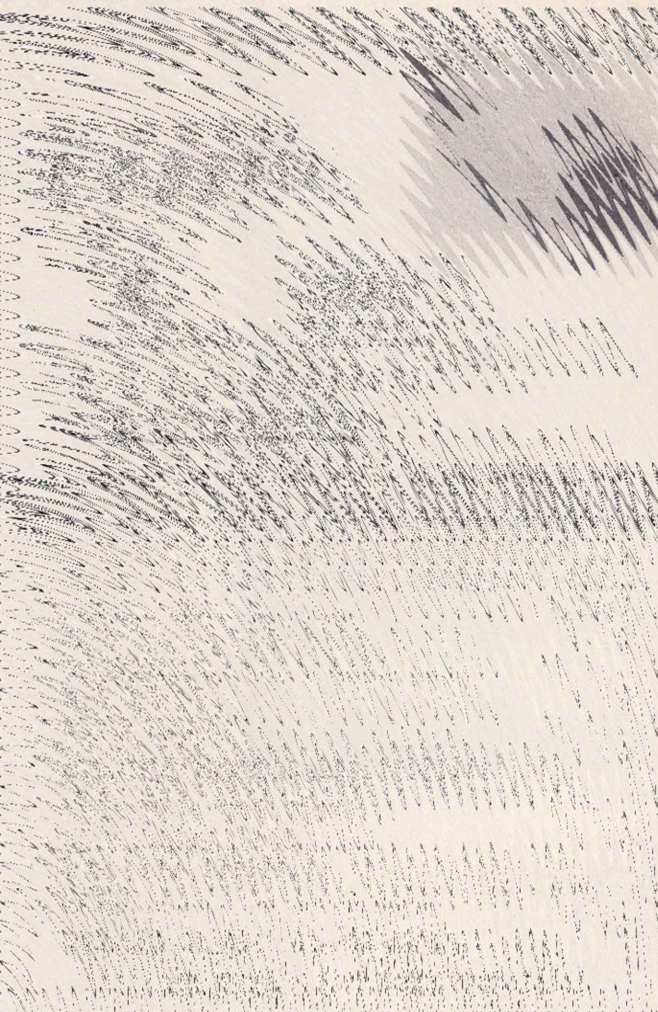


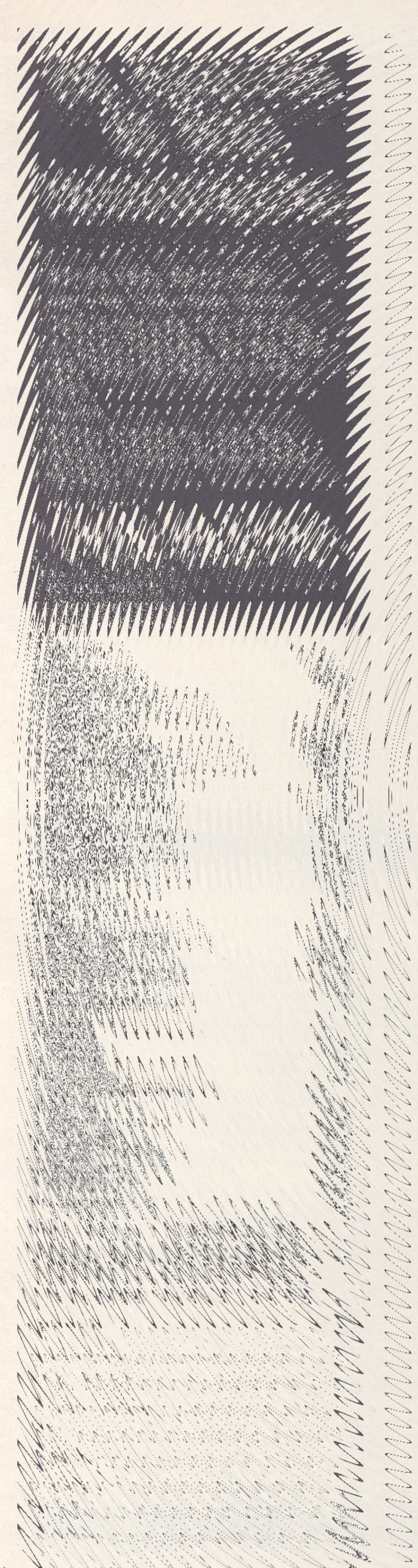
COMPUTER-MARKT

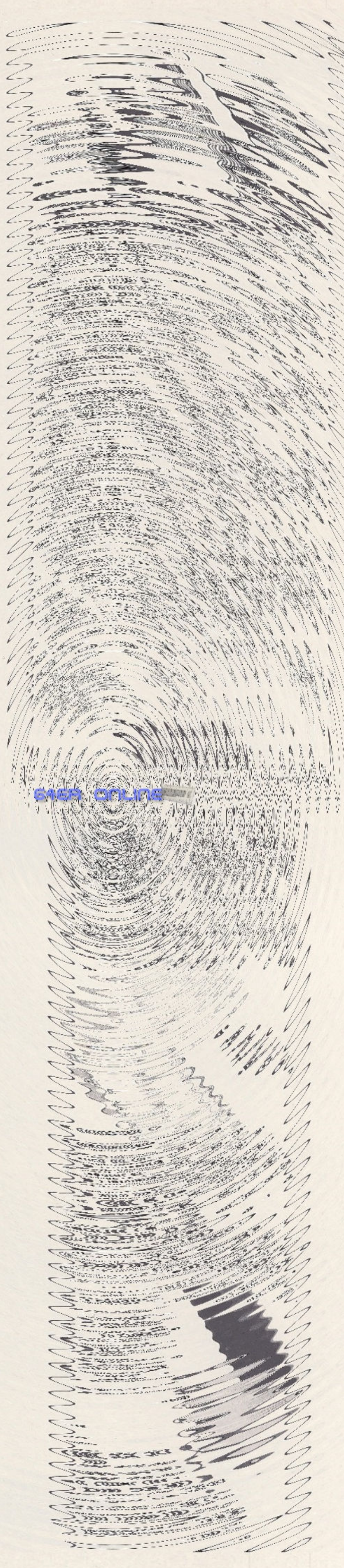
Private Kleinanzeigen Private Kleinanzeigen





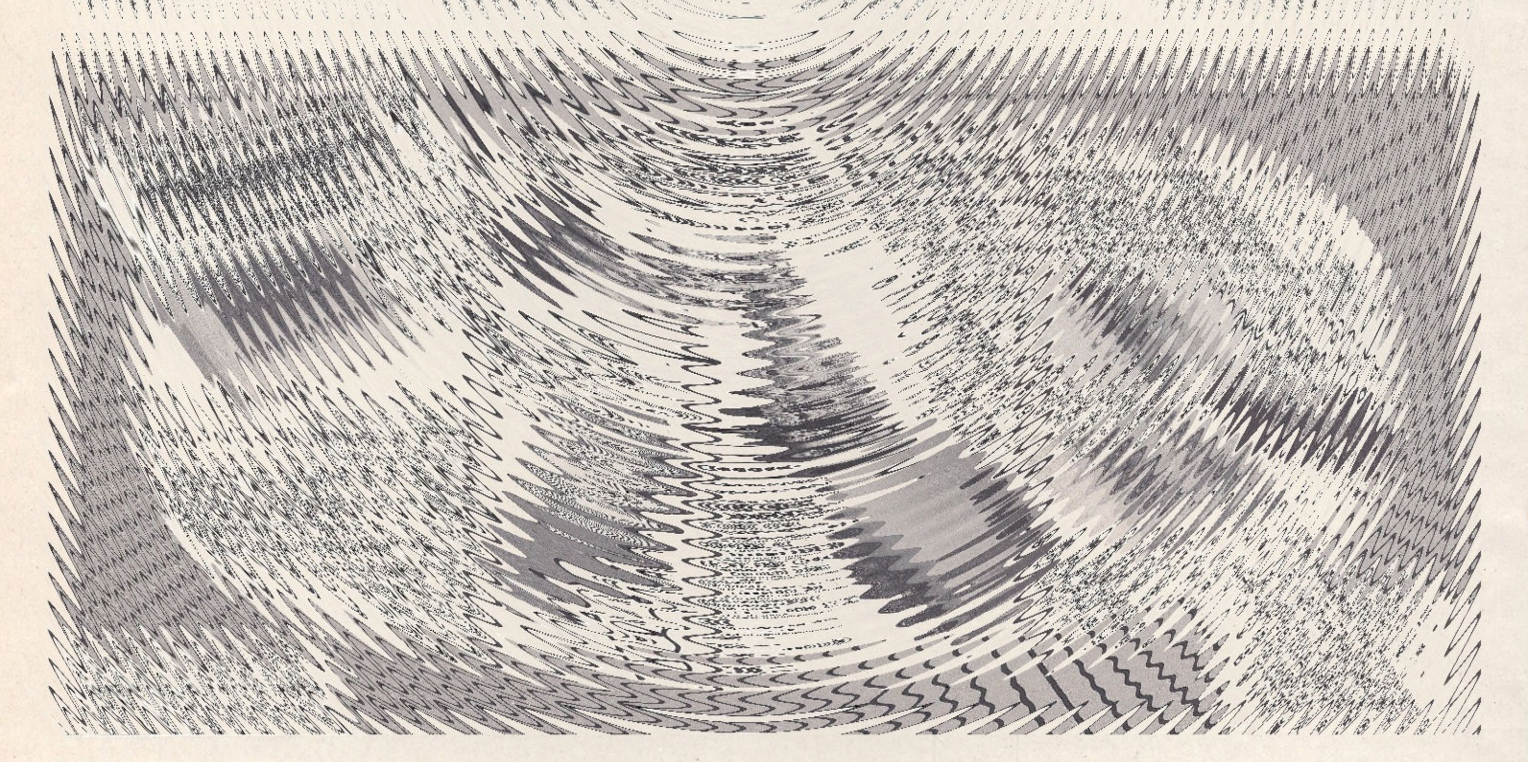




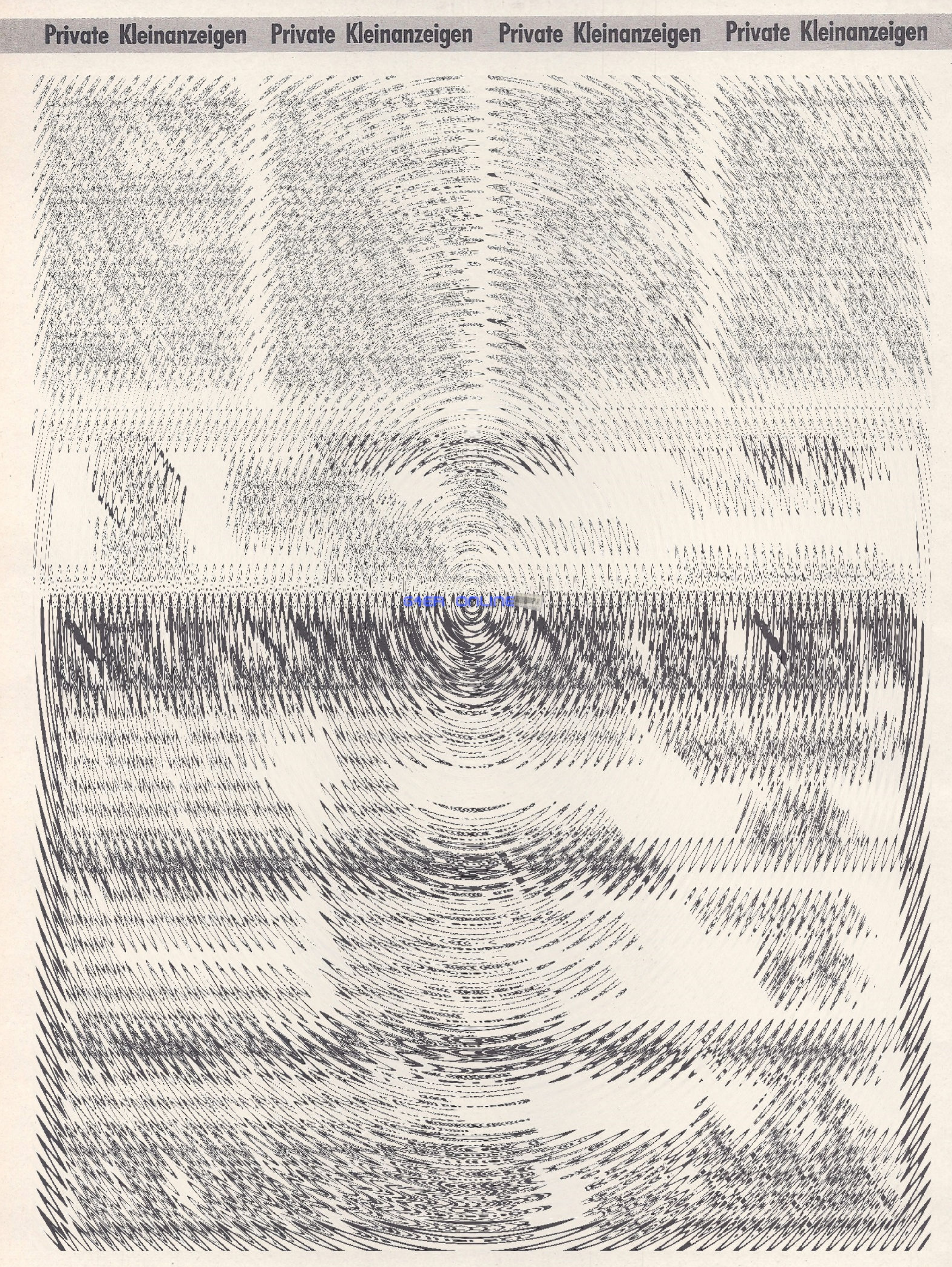


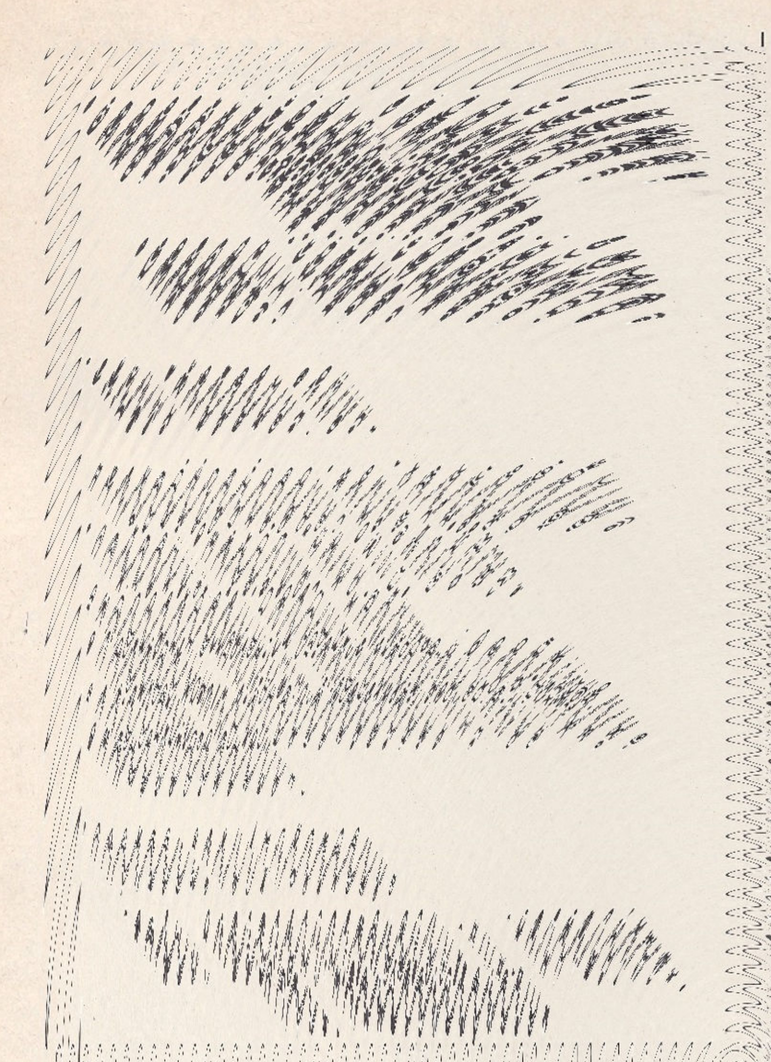
Private Kleinanzeigen

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen





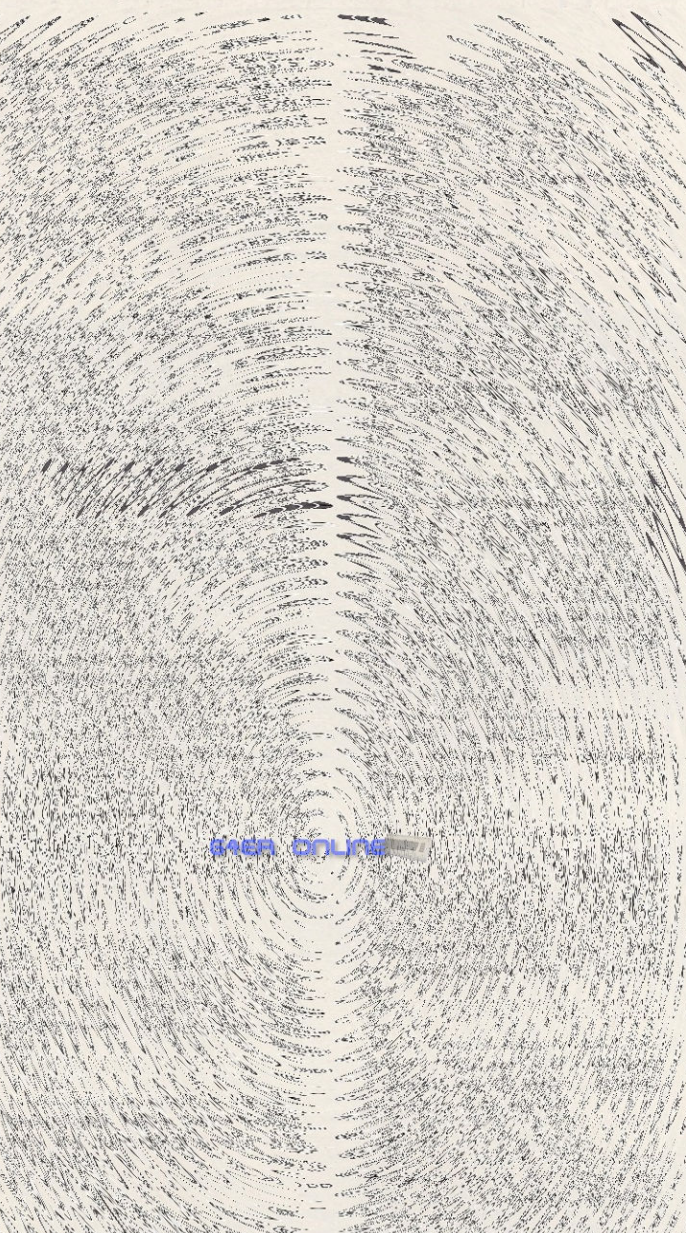


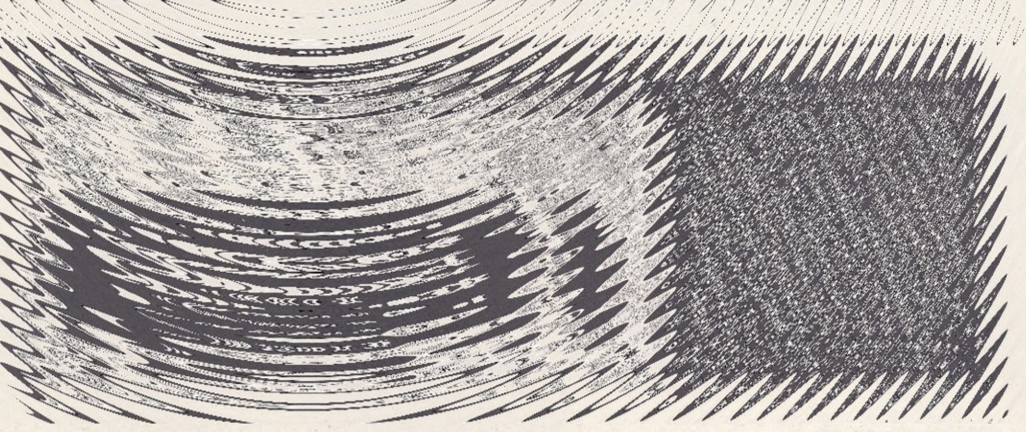




COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

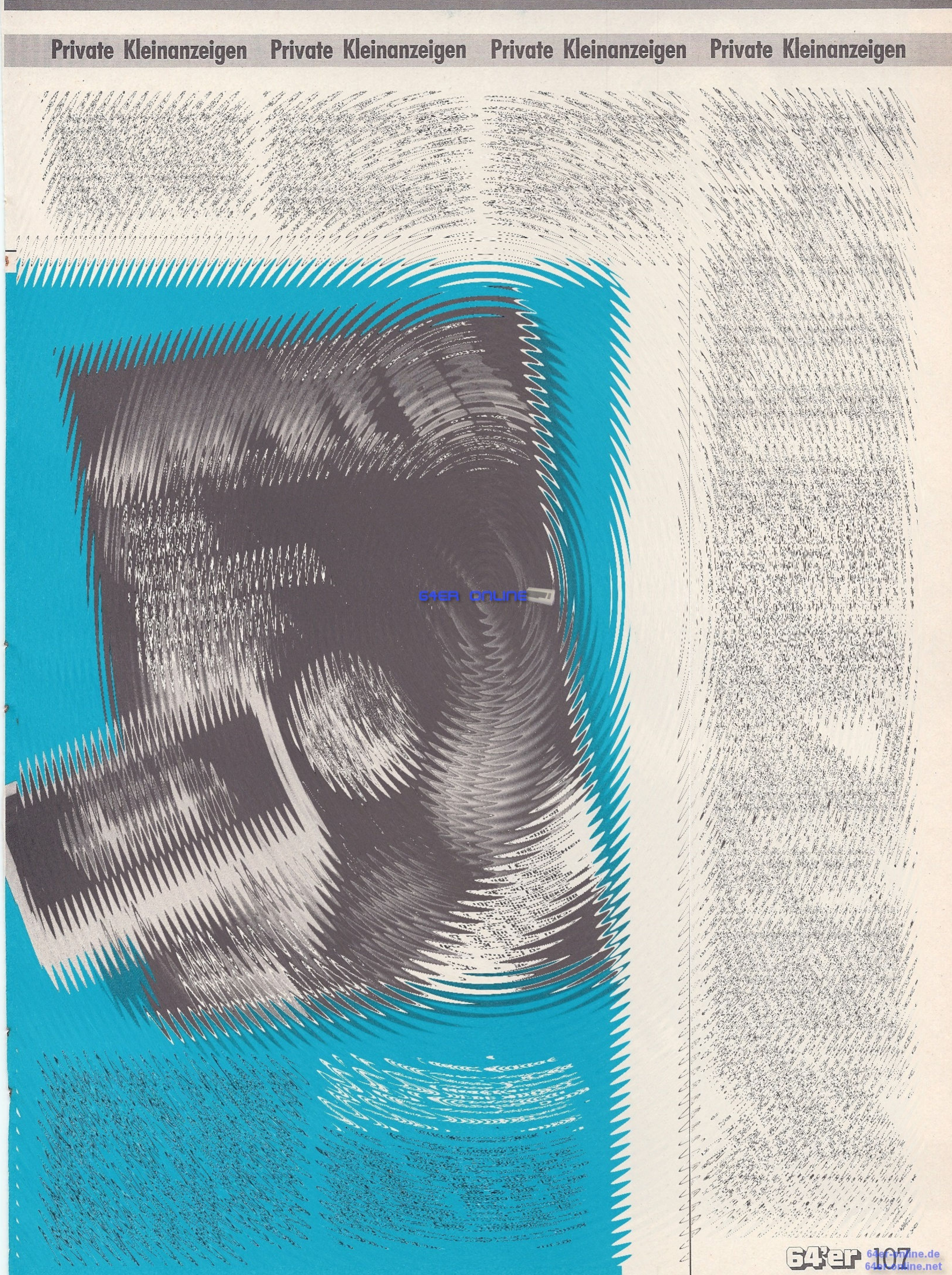




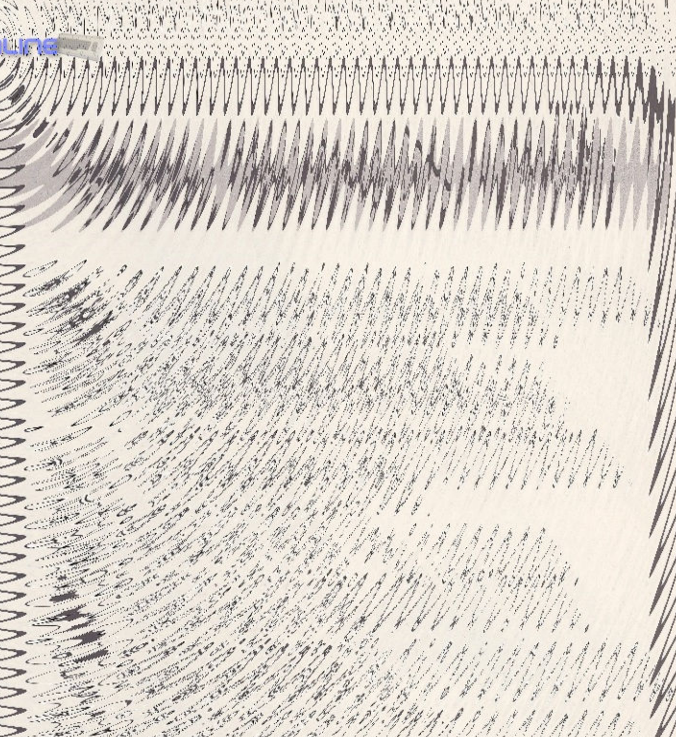
Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

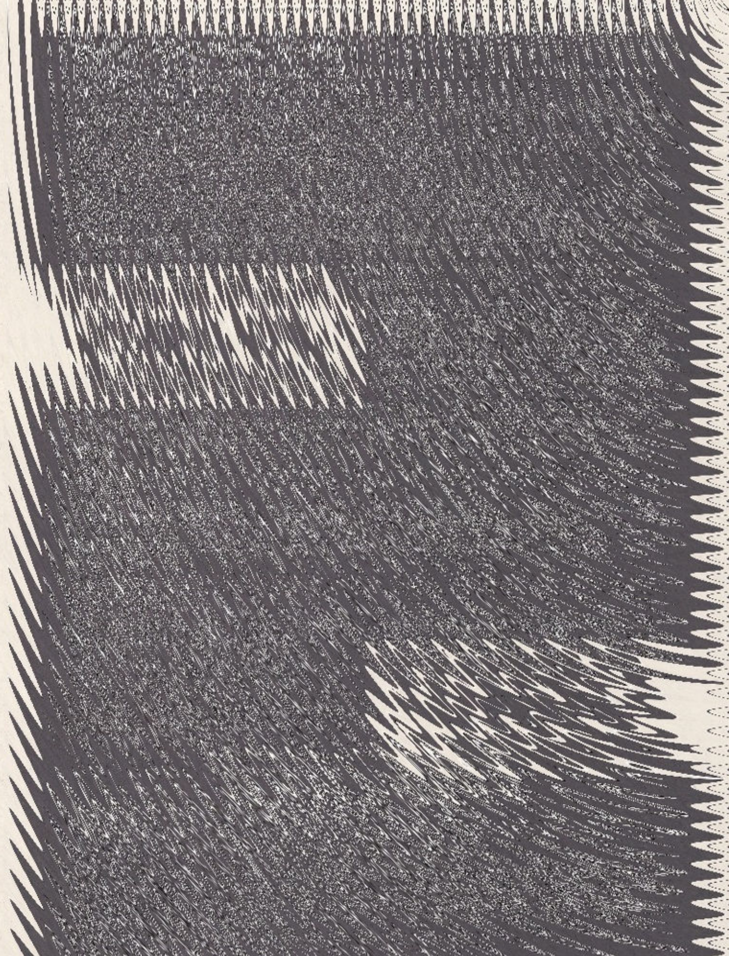


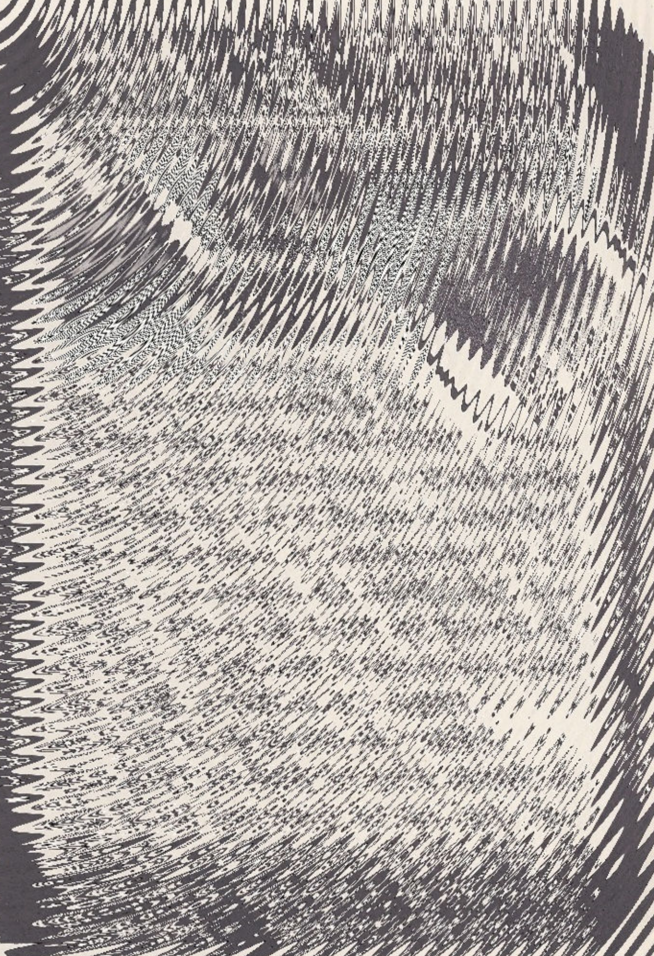


 Private Kleinanzeigen



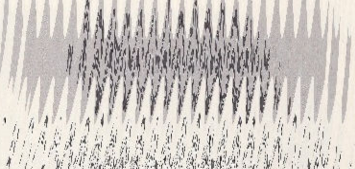






COMPUTER-MARKT

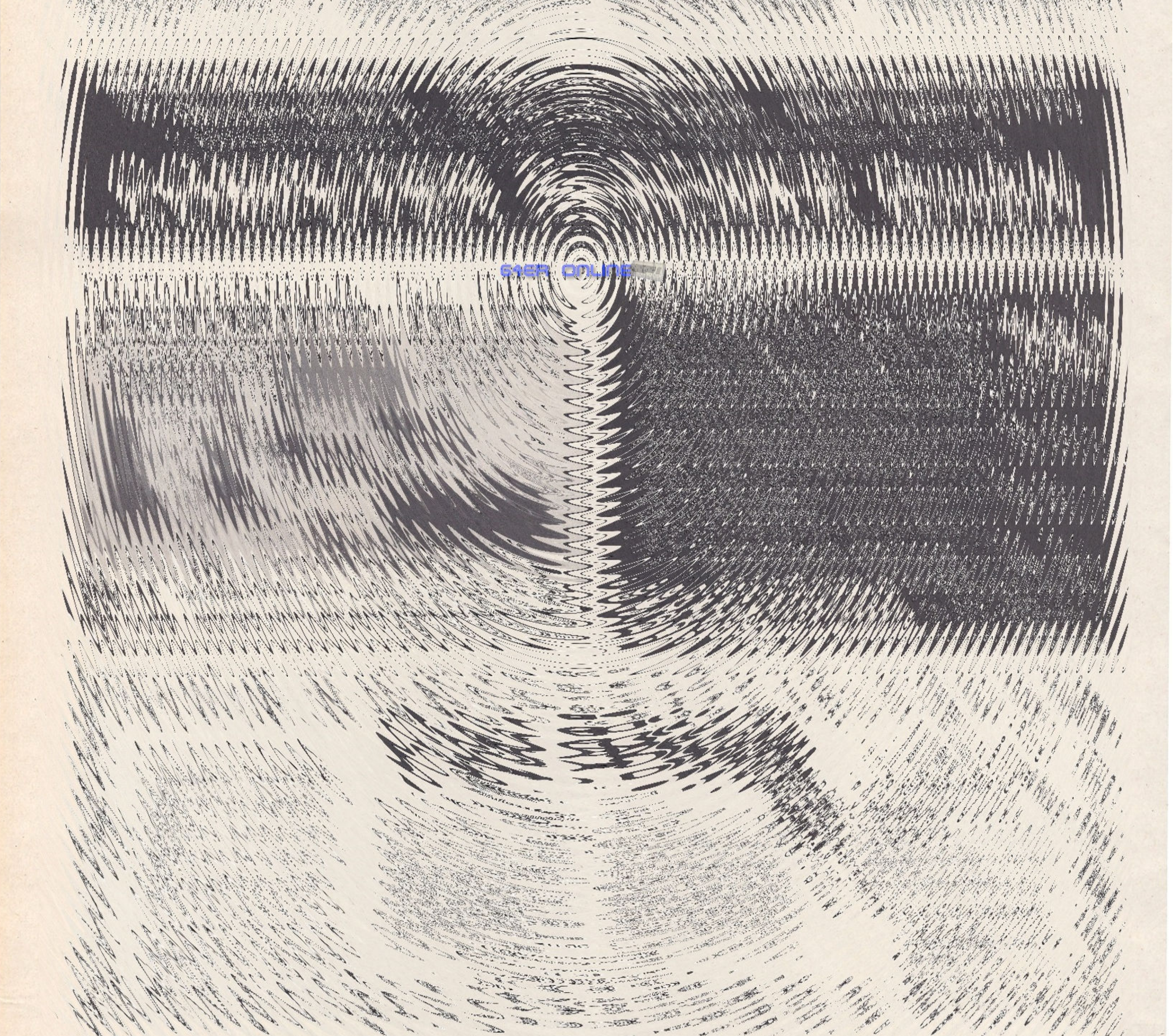
Private Kleinanzeigen Private Kleinanzeigen

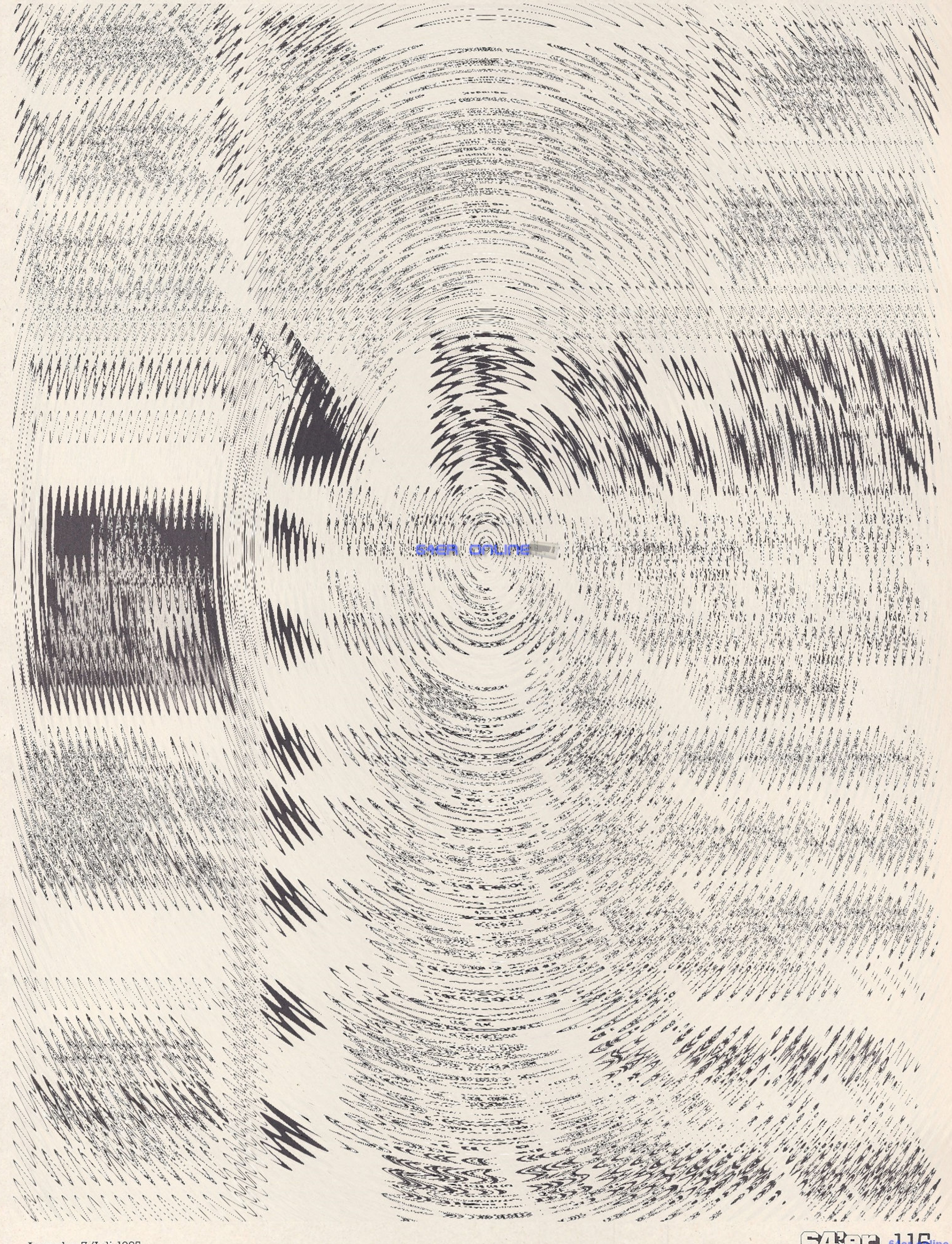


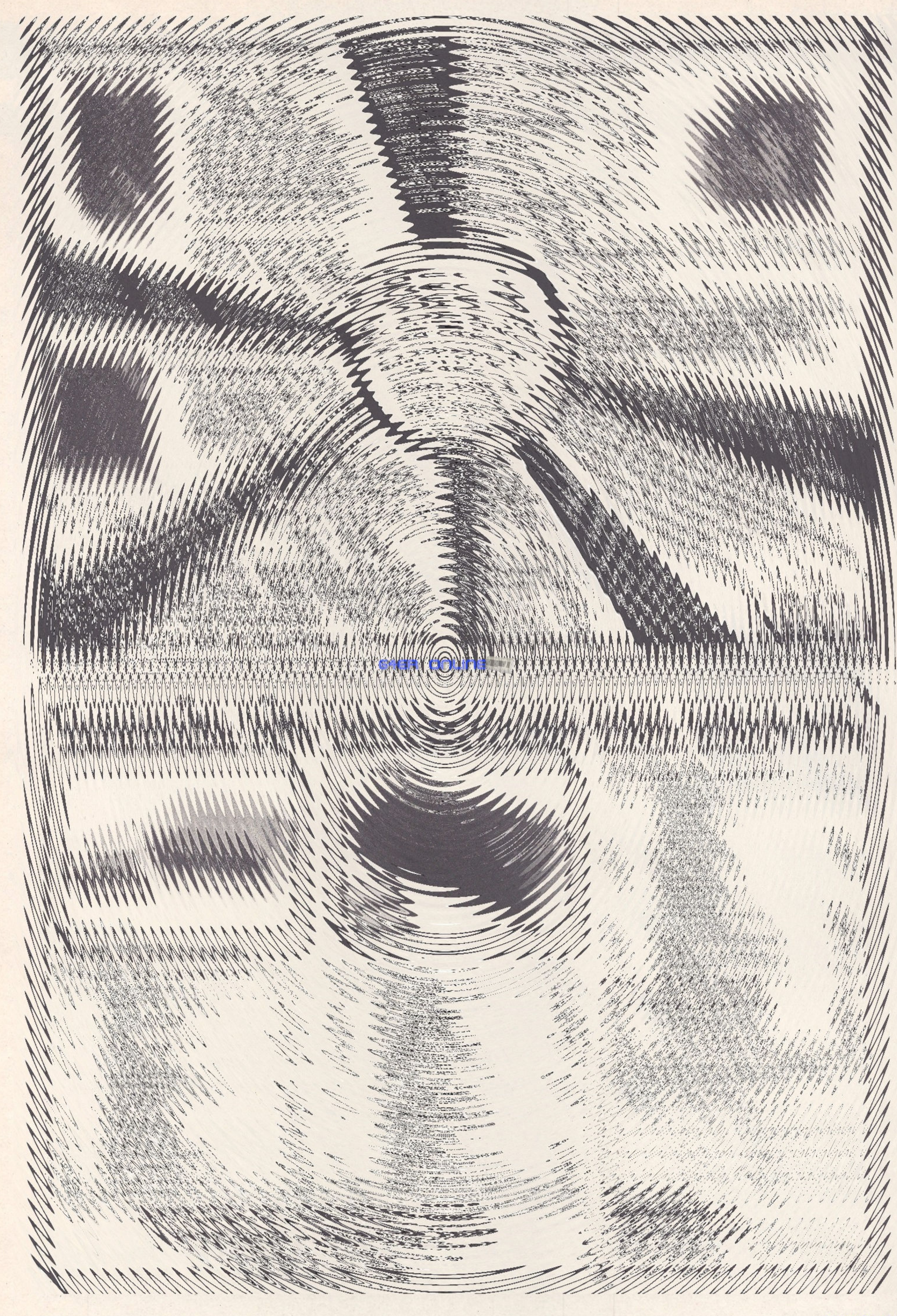
Gewerbliche Kleinanzeigen

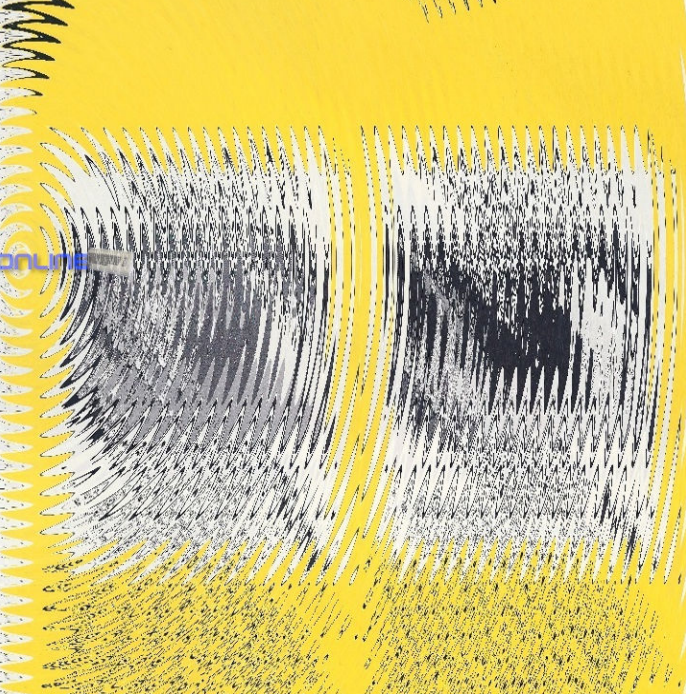
Gewerbliche Kleinanzeigen

Gewerbliche Kleinanzeigen



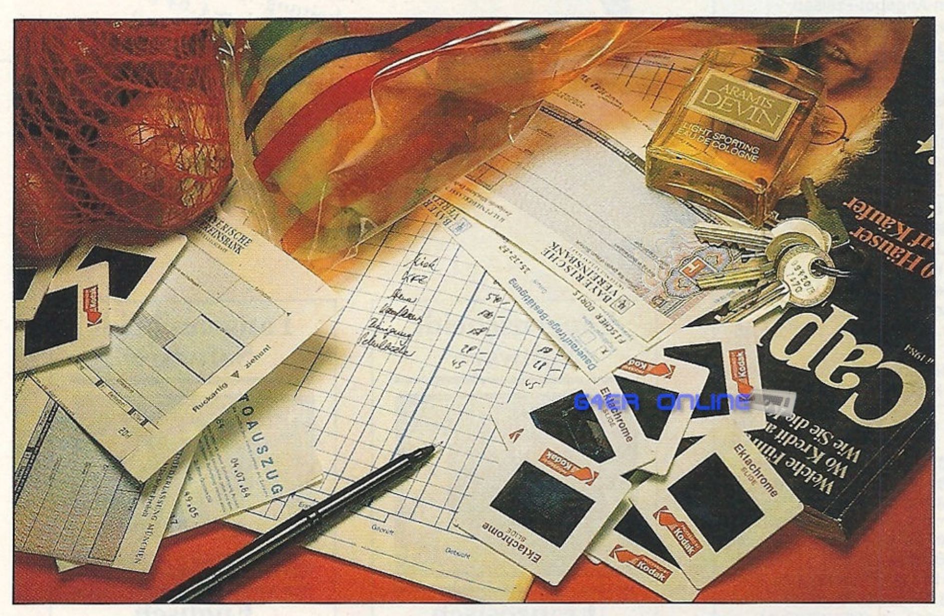






Vergleichstest: 7 Dateiverwaltungen auf einen Blick

Daten zu verwalten gehört neben der Textverarbeitung zum Hauptanwendungsgebiet eines Computers. Um Ihnen eine Übersicht zu vermitteln, stellen wir die sieben bekanntesten Programme vor.



Der Computer hilft bei der Verwaltung Ihrer Daten.

In diesem Kurzvergleich können unmöglich alle Einzelfunktionen sämtlicher Produkte aufgeführt werden. Einige Grundfunktionen (wie zum Beispiel das Eingeben, Ändern, Löschen und Ausgeben von Datensätzen) werden deshalb als selbstverständlich vorausgesetzt. Im wesentlichen wird jedoch gezielt auf solche Besonderheiten einzelner Produkte hingewiesen, die nicht gleichermaßen zum Standard gehören.

Datamat

Bereits im August '83 ist die erste Version von Datamat (aus dem Haus Data Becker) auf den Markt gekommen. Laut Data Becker soll dieses Programm bereits 15000 mal im Einsatz sein, was sicher auch auf den recht günstigen Preis zurückzuführen ist.

Die hier getestete neue Version (sie wird seit zirka August 1984 ausgeliefert) ist gegenüber der alten sowohl um einiges schneller geworden (durch Programmierung in Maschinensprache) als auch besser gegen Abstürze gesichert. Mit der alten Version erstellte Masken und Dateien können übernommen werden. Die Pointer-, Auswertungs- und Indexdatei müssen jedoch erneuert werden.

Durch ein gut gegliedertes deutsches Handbuch und eine komfortable Menüsteuerung wird auch der Einsteiger zum Erfolgserlebnis kommen (Bild 1).

Generell in allen Programmteilen gilt:
F1-Taste = Auslösetaste (wie RETURN)
F2-Taste = Abbruch (letztes Menü)
Leertaste = Springen im Menü

Die Erstellung von Masken ist einfach und problemlos. Die minimale Feldlänge von zwei Zeichen gestattet leider keine Definition von einstelligen Feldern. Dagegen ist es sehr angenehm, daß pro Satz bis zu 50 Felder definiert werden können. Die Anzahl der möglichen Datensätze hängt von der Länge des Indexfeldes und der Länge des einzelnen Satzes (maximal 2000) ab. Das Progamm bietet umfangreiche Suchfunktionen:
— Schnelles Suchen über das Indexfeld
— Suchen über Index mit Joker (Name=M*)
— Suchen nach beliebigen Kriterien

Bei der letzten Funktion sind mehrere Kriterien kombinierbar. Es kann auch jedem Suchfeld ein Vergleichsoperator wie gleich, kleiner, größer, kleiner gleich, größer gleich, ungleich zugeordnet werden. Interessant gelöst ist auch der Intervall-Vergleich von — bis, bei dem mit zwei Bildschirmebenen gearbeitet wird. Die ausgewählten Datensätze können mehrstufig umsortiert werden (Pointer-Datei als Hilfsdatei). Die Ergebnisse sind am Bildschirm leider nur einzeln abzurufen, können jedoch als frei definierbare Liste verdichtet auf Drucker ausgegeben werden. Diese Listengestaltung ist sehr komfortabel. Durch horizontales Verschieben wird das Listenformat auch am Bildschirm als ganzes erkennbar. Für jede beliebige Spalte kann außerdem eine Summenbildung durchgeführt werden.

Über die Menüpunkte »Datei auswerten« und »Floppy« kann eine Datei mit neuer Feldanordnung erzeugt werden, die dann mit einem Textverarbeitungsprogramm weiterverarbeitet werden kann. Wie bei der Textverarbeitung ist auch bereits im Datamat ein deutscher Zeichensatz wählbar.

Datenmanager 64

Die deutsche Beschreibung ist leider genauso dürftig wie man es von Commodore-Handbüchern eben gewohnt ist. Sie umfaßt ganze 19 Seiten. Zum Glück ist das Produkt selbst besser. Nach dem Start erscheint ein umfangreiches Menü (Bild 2). Die Auswahl erfolgt durch Eingabe des jeweiligen Anfangsbuchstabens. Auch ohne Menü kann im weiteren jederzeit ein Kommando direkt gegeben werden oder mit »H« (Help) zum Menü gesprungen werden.

Zum Aufbau einer Datei werden zunächst alle Felder (maximal 10!) des Datensatzes definiert: Feldname, Länge (maximal 79), Typ (a/n = alphanumerisch/numerisch), Index (j/n). Es sind maximal drei Indexfelder möglich. Auf die Feldanordnung

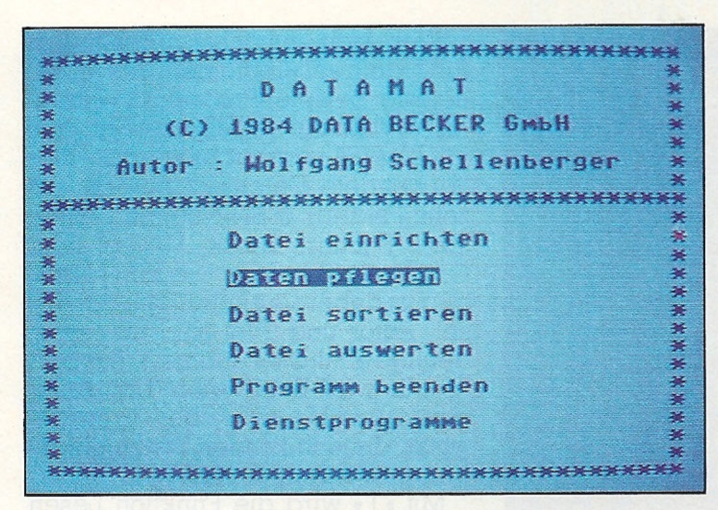


Bild 1. Eines der vielen Menüs des Datamat

Programmwahl
Dateiaufbau · Wechsel v. Disk
Eingabe · Information
Suchen · Help
Blaettern · Parameter
Aendern · Farbe
Loeschen · Textverarbeitg.
Geordnete Listen · Ready
Kommando ?

Bild 2. Das Hauptmenü des Datenmanagers

in der Eingabemaske hat man keinen weiteren Einfluß, sie erfolgt automatisch zeilenweise. Die Datei mit maximal 1000 Sätzen kann auch über mehrere Disketten verteilt werden.

Das Programm bietet neben dem Suchen über eines der Indexfelder auch die Möglichkeit, nach mehreren Kriterien auszuwählen (UND-Verknüpfung). Hinter dem Suchbegriff kann auch ein logischer Operator wie kleiner, größer, gleich angegeben werden. Eine Sortierung ist nur über eines der Indexfelder möglich. Durch entsprechende Voreinstellung im Menüpunkt »Parameter« ist auch die Ausgabe von geordneten Listen oder die Erstellung von Adreßaufklebern möglich. Interessant ist der Änderungsmodus für numerische Felder: Neben Überschreiben ist hier auch die Änderung um einen absoluten Wert oder eine prozentuale Änderung möglich, zum Beispiel Stück: +20 (Zubuchen zum Lagerbestand ohne Kopfrechnen) oder Preis: +3% (Erhöhung des gespeicherten Preises).

Der Datenmanager besitzt außerdem eine einfache, integrierte Textverarbeitung mit einer maximalen Zeilenlänge von 60 Zeichen und einer maximalen Textlänge von zirka vier Schreibmaschinenseiten. Mit » ' « können auch Felder zur Erstellung von Serienbriefen definiert werden.

Datei 64

Die Programmdiskette wird ohne Handbuch nur mit zwei Seiten Kurzinformation ausgeliefert. Nach dem Laden des Programms erscheint der sogenannte »Hauptverteiler«, ein auf die Funktionstasten abgestimmtes Menü (Bild 3). Mit der F1-Taste bekommt man eine äußerst knappe Erklärung der Programmbedienung. Zwei Maskendefinitionen werden bereits mitgeliefert, eine Anschriftenverwaltung und ein Bücherarchiv. Möchte man eine eigene Anwendung erstellen, so ist zunächst ein Datensatz zu definieren:

Feldnamen, Eingabeart (alpha, Betrag, Datum etc.) und Feldlänge. Bei Datums- und Uhrzeitfeldern wird sogar eine Plausibilitätskontrolle durchgeführt.

Danach muß für jedes Feld die genaue Position festgelegt werden. Außerdem können noch Überschriftsund Titelzeilen definiert werden. Bevor die Dateneingabe beginnt, wird eine Arbeitsdiskette erstellt.

Am Beispiel des Bücherarchivs sollen nun die Möglichkeiten dieses Programms kurz erläutert werden. Im Datensatz sind die Felder Buchtitel, Autor, Sachgebiet, Stichworte und Archivnummer gespeichert. Um einen Satz wiederzufinden, muß erst angegeben werden, nach welchem Feld gesucht werden soll, zum Beispiel (B) = Autor. In die dann erscheinende Zeile wird der Name eingegeben. Für die Suche kann auch ein Intervall von — bis vorgegeben werden. Die Ergebnisse lassen sich einzeln oder als Liste ausdrucken. Eine Gesamtliste kann auch nach einem bestimmten Feld sortiert werden. Für die Liste lassen sich bestimmte Felder unterdrücken, für jede Spalte ist auch eine Summenbildung möglich.

SM-Adreva/64

Das Programm wird mit einer kurzgefaßten deutschen Beschreibung, leider ohne Stichwortverzeichnis, ausgeliefert. Dafür wird jedoch neben der rein verbalen Beschreibung sehr schön in kleinen Ablaufplänen die logische Reihenfolge von Funktionstasten und Dateneingaben erläutert, so daß sich auch der Anfänger gut zurechtfindet. Aus dem Start-Menü lassen sich zunächst Commodore- oder deutsche Tastatur, die Druckparameter und die Farbeinstellungen wählen. Danach kann die Adreßeingabe beginnen.

Die fest definierte Maske (Bild 4) enthält neben den üblichen Adreßzeilen noch die Felder Kunden-Nummer (3-stellig), Suchbegriff (10-stellig), Telefon- und Telex-Nummer, sowie fünf weitere Zusatz-Textzeilen.

Am unteren Bildschirmrand ist während der gesamten Verarbeitung die jeweilige Funktionstastenbelegung eingeblendet, zum Beispiel speichern, löschen, drucken, Programmende.

Erfaßte Adressen können über die Kunden-Nummer oder über den Eintrag im Feld »Suchbegriff« wiedergefunden werden. Von einem gefundenen Satz kann man jeweils vor- und zurückblättern.

Der Ausdruck einer Adresse ist fest definiert und gibt jedes Feld in einer neuen Zeile aus. Standardmäßig können auch Adreßlisten oder Adreßaufkleber erstellt werden. Dabei ist die Auswahl über Kunden-Nummer (von-bis) oder Suchbegriff (von-bis) möglich.

Die Ausgabe erfolgt in der entsprechenden Sortierung.

SM-Adreva/64 kann kombiniert werden mit der Textverarbeitung SM-Text/64. Der mit der Textverarbeitung erstellte Brief kann dann Kürzel enthalten, für die die entsprechenden Daten aus der Adreßverwaltung übernommen werden. Damit lassen sich auf einfache Weise auch Form- und Serienbriefe erstellen.

SM-Adreva/64 ist ein Programm, das zwar speziell auf die Adreßverwaltung zugeschnitten ist, das aber dafür durch gute Benutzerführung besonders einfach zu bedienen ist.

Bereits beim Lesen des umfangreichen Handbuchs wird klar, daß

Bild 3. Datei 64 ist das preisgünstigste Programm

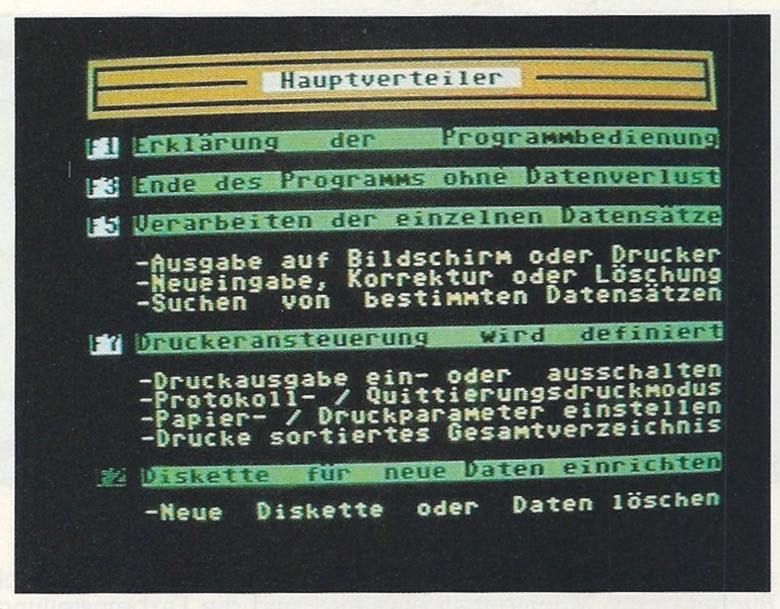


Bild 4. SM-Adreva beim Anzeigen eines Datensatzes. Auf ein Foto zu SM-ISM/64 wurde verzichtet, da der Bildschirmaufbau vom Benutzer selbst programmiert werden muß.

Bild 5. Superbase ist das wohl leistungsfähigste Dateiverwaltungsprogramm.

Bild 6. Vizastar besitzt sowohl Tabellenkalkulation mit Grafikteil als auch Dateiverwaltung.

das Produkt ISM/64 (Index Sequential File Manager) sich nicht ohne weiteres für den Anfänger eignet. Um all die vielfältigen Möglichkeiten zu nutzen, muß man schon tiefer einsteigen. Dem Profi dagegen wird bereits jetzt das Herz höher schlagen. ISM/64 ist vollständig in 6502-Maschinensprache geschrieben

SM-ISM/64

und läuft außer auf dem C 64 auch auf dem 8032 und 8096 von Commodore. Hier zunächst einige wesentliche Eigenschaften:

— variable oder feste Satzlänge von 2 bis 31875 Byte
— bis zu 125 Felder pro Satz
— Felder fester oder variabler Länge
— Feldlänge maximal 255 Byte
— maximal 40 Schlüsselfelder
— Schlüssel können auch nachträglich definiert oder gelöscht werden.
— Schlüssel können eindeutig oder mehrdeutig sein.
— maximal 10 ISM-Dateien können dem Programm gleichzeitig zur Verfügung gestellt werden.
— ISM kann auch in ein Basic-Programm eingebunden werden.

Insgesamt wird nach diesem Leistungsumfang deutlich, daß es sich bei ISM/64 bereits mehr um eine datenbankorientierte Programmiersprache handelt.

Wer sich zunächst überfordert fühlt, dem sei empfohlen, das mitgelieferte Beispiel-Programm »Literaturverwaltung« durchzuspielen, um mit ISM/64 vertraut zu werden. Der verwendete Datensatz besteht aus zehn Feldern: ISBN-Nummer, Autor, Titel, Buchart, Inhalt, Verlag, Jahr, Ort, Preis, Signatur. Die ISBN-Nummer ist dabei als eindeutiger Schlüssel, Autor und Titel sind als mehrdeutige Schlüssel definiert. Aus den Menüs wird Bearbeiten (bb), Anlegen (a) und Neuschreiben (n) ausgewählt, danach können Daten eingegeben werden.

Mit »1« wird die Funktion Lesen aufgerufen. Über den vollen oder verkürzten Schlüssel wird auf die eingegebenen Sätze zugegriffen.

Mit »z« kommt man jederzeit zum Hauptmenü zurück. Ein gefundener Satz kann direkt geändert oder gelöscht werden.

Alles in allem ist ISM/64 ein mächtiges Progammpaket, das in diesem Rahmen gar nicht in allen Einzelheiten aufgezeigt werden kann, das aber auch sehr hohe Anforderungen an den Anwender stellt, sofern ►


64er online
64er-online.de
64er-online.net

ket, das folgende drei Komponenten beinhaltet: Tabellenbearbeitung, Dateiverwaltungssystem und Grafikauswertung.

Im Rahmen dieses Vergleichs wird vor allem die Datenbankverwaltung genauer untersucht. Eine Vizastar-Datenbank ist eine Sammlung von bis zu 15 Dateien. Ein Datensatz kann bis zu 8000 Zeichen beinhalten und sich über neun »Bildschirmkarten« erstrecken. Die maximale Feldlänge beträgt 120 Zeichen.

Alle Vorgänge sind menügesteuert (SPACE-Taste zum Weiterspringen, mit RETURN ausführen). Mit der EXEC-Funktion können aber auch ganze Befehlssequenzen vorprogrammiert werden.

Vizastar 64

Die Datensatzfelder werden im elektronischen Arbeitsblatt festgelegt, jedem Feld wird ein Buchstabencode zugeordnet (A-BL). Es kann ein Schlüsselfeld mit 30 Zeichen Länge definiert werden, außerdem Formelfelder und beliebige Datenfelder (maximal 120 Zeichen). Mit ADD, REPLACE und DELETE werden Datensätze eingegeben, geändert oder gelöscht. Der Zugriff erfolgt über den Schlüssel (KEY) oder über andere Kriterien (DATA USE CRITERIA), die einfach in die Felder eingegeben werden (inklusive Vergleichsoperatoren oder Joker). Mit dem Befehl DATA TRANSFER lassen sich Daten zwischen elektronischem Arbeitsblatt und Datensatz austauschen. Mit EXPORT können Daten an andere Programme wie zum Beispiel die Textverarbeitung Vizawrite 64 übergeben werden, umgekehrt mit IMPORT. Mit REPORT und RANGE lassen sich beliebige Felder für den Listenausdruck zusammenstellen.

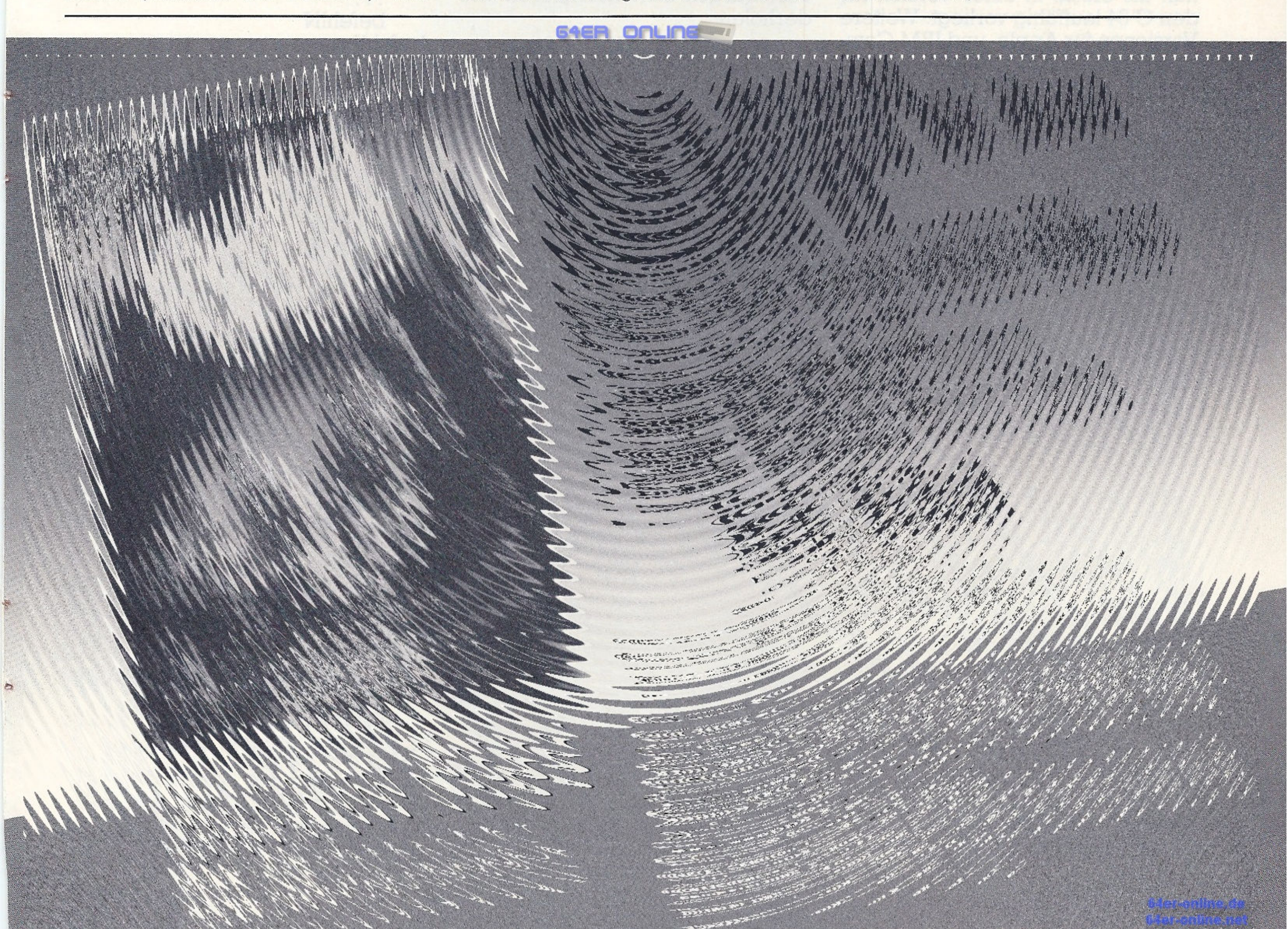
Neben der Datenbank-Komponente verfügt Vizastar 64 über eine sehr leistungsfähige Tabellenbearbeitung inklusive Invertierung und Window-Technik. Der Grafikteil umfaßt standardmäßig Balken- und Liniendiagramme. Mit einem Grafik-Zusatzprogramm sind farbige Kuchengrafik und 3D-Balkengrafik möglich.

Insgesamt bietet Vizastar 64 eine Vielzahl von Möglichkeiten mit angenehmer Benutzerführung in allen Programmteilen.

Ein derartiger Vergleich unterschiedlichster Datenbanksysteme kann als Ergebnis sicher keinen eindeutigen Favoriten oder Verlierer präsentieren. Er kann nur gleichzeitig über mehrere Produkte Informationen liefern, die dem einzelnen Interessenten die Möglichkeit geben, systematisch zu untersuchen, was für seinen speziellen Anwendungsfall überhaupt in Frage kommt. Gewisse Mindestvoraussetzungen müssen sicher erfüllt sein, damit sich ein Problem mit einem dieser Produkte überhaupt lösen läßt, daneben sollte natürlich ein vernünftiges Preis-Leistungsverhältnis Berücksichtigung finden. Weitere Angaben entnehmen Sie bitte der Tabelle.

(Peter Häuslein/gk)

Bezugsquellennachweis:
Commodore, Lyonerstr.38, 6000 Frankfurt 71 (Datenmanager)
Data Becker, Merowingerstr.30,4000 Düsseldorf 1 (Datamat, Superbase)
Dynamics, Große Bäckerstr. 11, 2000 Hamburg 1 (Datei 64)
Interface Age, Josefsburgstr.6, 8000 München 80 (Vizastar)
für Schweiz: Microtron, Brunnenweg 5, CH-2542 Pieterlen (Vizastar)
SM Software AG, Scherbaumstr.33, 8000 München 8 (SM-Adreva, SM-ISM)



Promal — Die neue Sprache für Profis?

Wir testeten eine brandneue Programmiersprache für den Commodore 64. Promal kommt direkt aus Amerika und ist schnell wie Forth, aber übersichtlich wie Pascal.

Die Programmiersprachen für den Commodore 64 häufen sich, und so vergeht kaum ein Monat, an dem nicht irgendein Hersteller ein neues Produkt als »Die Sprache für den Commodore 64« anpreist. Meistens handelt es sich dann doch aber nur um mehr schlechte als rechte Implementationen altbekannter Sprachen, die vielleicht noch auf die Grafik- und Soundfähigkeiten des C 64 Rücksicht nehmen.

Ein ganz neues Produkt ist dagegen die Sprache Promal, die erst letztes Jahr entwickelt wurde und nun als erstes in einer Version für den C 64 lauffähig vorliegt. Weitere Versionen für Apple- und IBM-Computer sind in Arbeit. Da Promal-Programme auch zwischen verschiedenen Computern austauschbar sind, solange nicht auf die Maschinenebene zugegriffen wird, dürfte daher bald ein breites Software-Angebot zur Verfügung stehen.

Was ist Promal?

Promal ist die Abkürzung für **Pro**grammers **M**icro **A**pplications **L**anguage und hat überhaupt nichts mit Comal zu tun. Promal bietet die Möglichkeit, strukturiert zu programmieren, ja erzwingt dieses ähnlich wie Pascal sogar, ist dabei aber gleichzeitig so schnell und maschinennah wie Forth.

Promal besteht aus drei Teilen: Dem Executive, dem Editor und dem Compiler. Der Executive ist ähnlich einem Interpreter organisiert und enthält den Kern von Promal sowie die Input/Output-Routinen und die Befehle, die direkt ausgeführt werden können. Über den Executive tritt der Benutzer mit dem Promal-System in Verbindung. Editor und Compiler werden vom Executive aus aufgerufen. Eine Programmierung in Promal ist über den Executive nicht möglich, wohl aber lassen sich bereits compilierte Programme starten. Es können mehrere Programme gleichzeitig im Speicher stehen: Promal-Programme sind vollkommen relokatibel (im Speicher verschiebbar) und können so jederzeit zu den schon im Speicher stehenden Programmen hinzugeladen werden. Der Aufruf eines Programmes erfolgt einfach durch die Nennung seines Namens. Befindet sich das entsprechende Programm nicht im Speicher, wird es automatisch nachgeladen.

Die Bedienung des Executive ist äußerst komfortabel: So sind zum Beispiel die Funktionstasten belegt, und es lassen sich schon einmal gemachte Befehls-Eingaben mittels Tastendruck zum Korrigieren wiederholen.

Komfortabler Editor

Soll nun ein neues Promal-Programm geschrieben werden, dann wird zunächst der Editor aufgerufen. Es handelt sich dabei schon fast um ein kleines Textsystem mit horizontalem Scrolling (80 Zeichen pro Zeile), schnellem vertikalem Scrolling in beide Richtungen, Funktionstastenbelegung, ausführlichem Befehlssatz und vielen anderen Details. Hat man eine Weile mit diesem schnellen und komfortabeln Editor gearbeitet, so ist man recht erstaunt, wenn man im Handbuch liest, daß er vollständig in Promal geschrieben wurde. Der Editor ist somit eines der besten Beispiele für die Leistungsfähigkeit von Promal.

Ebenso in Promal geschrieben ist der Compiler. Es kann wahlweise auf Diskette oder im Speicher compiliert werden. Auch hier erstaunte die enorme Geschwindigkeit, zumindest solange keine Diskettenzugriffe notwendig sind.

Wie sieht nun so ein Promal-Programm aus? Auf den ersten Blick glaubt man, es mit einer Mischung aus Pascal und Basic zu tun zu haben. Es gibt keinerlei Zeilennummern, aber Einrückungen. In jeder Zeile steht nur ein einziger Befehl. Zu Anfang eines jeden Promal-Programmes steht ein Deklarationsteil, in dem alle Variablen, Funktionen und Prozeduren (Unterprogramme) ähnlich wie in Pascal definiert werden müssen.

Es gibt sehr viele veschiedene Kontrollstrukturen: Neben dem obligatorischen IF ... ELSE (ohne THEN!) oder FOR (ohne NEXT) gibt es auch WHILE, REPEAT...UNTIL, und sogar ein CHOOSE, das in etwa dem ON...GOTO (Basic) oder dem CASE (Pascal) entspricht, aber noch flexibler ist.

Bitte einrücken!

Wer sich fragt, wo das NEXT oder das THEN hin verschwunden sind: Promal zwingt zu strukturiertem Programmieren, weil Einrückungen syntaktische Bedeutung haben! Eine IF..THEN..ELSE-Abfrage muß in Promal folgendermaßen programmiert werden:

```
IF <Bedingung>
  befehl1
  befehl2
  ...
  befehlN
ELSE
  befehl1
  befehl2
  ...
  befehlN
befehl1
befehl2
...
```

Wichtig ist, daß die Befehlsgruppen, die bedingt ausgeführt werden sollen, um zwei Stellen eingerückt werden. Der Promal-Compiler nimmt an, daß die IF...ELSE-Anweisung zu Ende ist, wenn Befehle nicht mehr eingerückt sind.

Dieses Prinzip gilt auch für alle anderen Kontrollstrukturen. Die aus Basic bekannten FOR-NEXT-Schleifen benötigen in Promal kein NEXT mehr, das Schleifenende wird am Fehlen der Einrückung erkannt. Promal kennt allerdings einen Befehl NEXT, der aber eine andere Bedeutung hat. Mit ihm kann an den Anfang der gerade bearbeiteten Schleife zurückgesprungen werden. Dies funktioniert nur bei REPEAT...UNTIL und WHILE, nicht bei FOR. Das genaue Gegenteil von NEXT ist BREAK. Mit diesem Befehl wird die gerade bearbeitete Kontrollstruktur ordnungsgemäß abgeschlossen und verlassen, es erfolgt sozusagen ein GOTO zur nächsten nicht eingerückten Zeile.

Natürlich können Kontrollstrukturen auch beliebig ineinander geschachtelt werden, dann sind halt mehr Einrückungen notwendig.

Promal kennt vier verschiedene Variablentypen: BYTE, INTeger,

WORD und REAL. String-Variablen sind nicht vorgesehen, man muß sich, ähnlich wie in Forth, mit Adreßvariablen behelfen. Das bedeutet, man legt einen String als Bytefolge irgendwo im Speicher ab und weist die Anfangsadresse des Strings einer Variablen zu. Über diese Variable kann man dann im weiteren jederzeit auf den String zugreifen.

Das mag auf den ersten Blick etwas umständlich erscheinen, ist aber in der Praxis sehr flexibel und schnell, insbesondere, da es von einigen Promal-Standardprozeduren unterstüzt wird.

An Feldvariablen sind nur eindimensionale Arrays erlaubt. Doch damit läßt sich bereits eine ganze Menge anfangen. Arrays können nämlich an jedem vom Programmierer vorgesehenen Speicherplatz abgelegt werden. Beispielsweise kann man ein solches Array auf den Bildschirm-Speicherbereich legen. Durch einfache Zuweisung von Werten an Array-Elemente wird dann jedesmal die entsprechende Speicherstelle im Bildschirm angesprochen. Dabei ergibt sich neben einer besseren Übersichtlichkeit des Programmes auch noch ein Zeitgewinn.

Verbesserte Fließkomma-Arithmetik

Im Gegensatz zu vielen anderen Programmiersprachen für den C 64 wurde bei der Fließkomma-Arithmetik in Promal nicht auf die bereits im Basic-ROM vorhandenen Routinen zurückgegriffen. Promal verfügt über eine völlig neue Fließkomma-Arithmetik, die die des Commodore 64-Basic bei weitem übertrifft. REAL-Variablen haben 11 signifikante Stellen (9 bei Basic), dennoch erfolgen alle Berechnungen damit fast doppelt so schnell. Damit ist die reine Ausführungszeit der Rechenoperation gemeint. Da Promal-Programme immer compiliert werden, ist die tatsächliche Geschwindigkeit von Programmen noch um ein Vielfaches höher.

Allerdings kennt Promal nur die vier Grundrechenarten und logische Operationen. Funktionen wie Sinus oder Logarithmus werden als Promal-Source-Code mitgeliefert. Nach der Compilation sind diese in Promal geschriebenen Routinen immer noch genauer und sogar ein wenig schneller als die Original-Commodore-ROM-Routinen, was wohl ein beeindruckender Beweis der Leistungsfähigkeit von Promal ist.

Wer in einem Programm ohne REAL-Zahlen auskommt, kann die REAL-Arithmetik vollständig aus Promal entfernen und gewinnt so zusätzlich wertvollen Speicherplatz. Compilate (übersetzte Programme) können so einen Umfang von bis zu 35 KByte haben. Der Source-Code muß dann allerdings mit einigen Tricks geschrieben werden, da der garantiert nicht in den Speicher paßt. Der Compiler selbst umfaßt beispielsweise als Sorce-Code 2400 Zeilen, benötigt aber compiliert nur noch 13 KByte.

Überzeugend ist auch das Konzept der Prozeduren und Funktionen. Pascal-Kenner wissen, daß Prozeduren das Gegenstück zu Unterprogrammen sind, und daß Funktionen ein spezieller Typ von Prozeduren sind, die Parameter an das Hauptprogramm zurückgeben. Wie in Pascal können Prozeduren und Funktionen lokale Variable haben, das sind Variable, die nur der Prozedur bekannt sind, und deswegen gleiche Namen wie völlig andere Variablen im Hauptprogramm haben dürfen, ohne daß Konflikte oder Fehler auftreten.

In Promal gibt es noch eine Unterscheidung bei lokalen Variablen: Diese können bei Bedarf ihren Inhalt bis zum nächsten Aufruf der Prozedur behalten. Eine Prozedur kann also beispielsweise zählen, wie oft sie hintereinander aufgerufen wurde. Dies ist sehr wichtig, will man rekursive Funktionen programmieren. Rekursion bedeutet, daß eine Prozedur oder Funktion sich selbst aufruft und ist in Promal beliebig erlaubt. Allerdings ist die Rekursionstiefe auf zirka 40 Aufrufe beschränkt, sonst läuft der Prozessorstack über, und das System befindet sich in einem »undefinierten Zustand«, etwas weniger fein auch als »Absturz« bezeichnet.

Schnittstelle zur Maschinensprache

Obwohl Promal bereits unglaublich schnell ist, hat man trotzdem mehrere Schnittstellen zur Maschinensprache eingebaut, um eigene Maschinenroutinen oder solche aus dem Basic-ROM in Promal-Programme zu integrieren. Dabei ist sogar die Kontrolle über die Prozessorregister (A, X, Y und P) von Promal aus vorgesehen. Parameterübergabe kann über gemeinsam benutzte Speicherstellen, die Prozessorregister oder am aller einfachsten über den Prozessorstack erfolgen. Damit ist Promal fast genauso maschinennah wie Forth, aber dafür um einiges komfortabler.

Die drei mitgelieferten Handbücher, wie üblich in Englisch, umfassen 250 DIN A4-Seiten und bieten eine umfassende Einführung in die Sprache. Da hier sehr viel mit Beispielen gearbeitet wird, fällt es leicht, sich in das Promal-Konzept einzuarbeiten.

Vorbildliche Dokumentation

Eines sei noch zur Erhältlichkeit von Promal gesagt: Es gibt drei verschiedene Promal-Versionen, die aber aufwärtskompatibel sind. Die Demo-Version umfaßt den kompletten Sprachschatz und unterscheidet sich von der End-User-Version nur dadurch, daß sie keine Druckerausgabe und kein Abspeichern von Programmen auf Diskette erlaubt. Sie ist also sozusagen nur zum »Reinschnuppern« gedacht. Dafür darf, ja soll sie sogar kopiert und an Freunde weitergegeben werden. Wer interessiert ist, kann sich dann die End-User-Version beim Hersteller kaufen und mit dem Programmieren voll loslegen. Dieses Konzept kann man anderen Anbietern von Software sicher nur zur Nachahmung empfehlen, da der Anwender die oftmals recht teuren Programme dann nicht mehr »auf Verdacht« kaufen muß, um erst hinterher enttäuscht festzustellen, daß das Programm nicht seinen Erwartungen entspricht. Für die Produzenten wirklich guter Software hätte dieses Verfahren ebenfalls nur Vorteile — der Bekanntheitsgrad erhöht sich, der Anwender kann sich ein Bild von der Qualität der Software machen.

Doch wieder zurück zu Promal. Die geschriebenen und compilierten Programme benötigen immer noch den Promal-Executive-Teil, um lauffähig zu sein. Deswegen gibt es noch eine »Developer's-Version«, die zusätzlich den Promal-Kern in die übersetzten Programme einbindet, so daß sie auch ohne Promal laufen können.

Zum erstenmal gibt es mit Promal eine maschinennahe und dennoch einfache und komfortable Sprache für Mikrocomputer. Jedem, dem Forth zu unverständlich und Pascal zu langsam ist, kann Promal nur empfohlen werden, denn diese Sprache verbindet sozusagen diese zwei Programmierwelten miteinander. (Boris Schneider/ev)

Info: Promal ist in den verschiedenen Versionen erhältlich bei Systems Management Associates, 3700 Computer Drive, P.O.Box 20023, Raleigh, North Carolina N.C. 27619, USA.

Forth-wärts mit M&T-Forth 64

Inzwischen gibt es fast schon so viele Forth-Versionen für den Commodore 64 wie Basic-Erweiterungen. Gerade neu auf dem Markt ist eine Version, die sich strikt an den FIG-Standard hält und damit völlige Software-Kompatibilität verspricht.

Forth hat in der letzten Zeit einen enormen Bekanntheitsgrad erreicht. Was vor einem Jahr noch ein Geheimtip unter Insidern war, liegt jetzt von den vielen Herstellern in den verschiedensten Ausführungen vor. Das grundlegende Konzept der Programmiersprache Forth ist von uns schon in früheren Ausgaben erläutert worden, daher seien an dieser Stelle nur noch einmal die wichtigsten Unterschiede zu Basic stichwortartig angesprochen:

Forth ist eine sehr maschinennahe Sprache, deren grundlegendes Prinzip der Stack ist. Alle Berechnungen werden grundsätzlich in der Weise ausgeführt, daß zunächst die Operanden auf den Rechenstack gebracht werden und anschließend erst die gewünschte Operation ausgeführt wird (ähnlich wie bei Taschenrechnern mit der »umgekehrten polnischen Notation« UPN). Eine Addition schreibt sich damit so:

5 7 +

Programmierung in Forth besteht darin, neue Worte zu definieren, die bestimmte Funktionen ausführen. Beispiel:

ADD1 1 +

Eine Wortdefinition beginnt mit einem Doppelpunkt und endet mit einem Semikolon. Hinter dem Doppelpunkt folgt der Name des neuen Wortes, im Beispiel also »ADD1«. Anschließend folgen die Befehle (Worte) die beim Aufruf von ADD1 ausgeführt werden sollen. ADD1 schiebt also die Zahl Eins auf den Stack und führt anschließend eine Addition durch. Im Endeffekt wird also der vor dem Aufruf von ADD1 auf dem Stack befindliche Zahlenwert um eins erhöht. In analoger Weise kann man ganze Programmsysteme aus immer leistungsfähigeren Forth-Worten aufbauen. Forth arbeitet ausschließlich mit Integer-Zahlen. Das führt im Zusammenhang mit der Compilation aller Eingaben zu sehr effizienten Programmen.

Das neueste Kind auf dem Forth-Markt ist eine Version der Markt & Technik Verlags AG. M&T-Forth 64 hält sich, soviel sei hier schon gesagt, strikt an den Semi-Standard der Forth-Interest-Group (FIG). Natürlich finden sich hier sehr viele Befehle, die auf den C 64 zugeschnitten worden sind. Insgesamt umfaßt das Grundvokabular von M&T-Forth 64 rund 280 Worte. Dazu kommen dann einzelne auf der Programmdiskette mitgelieferte Programme, die ja bei Forth nichts anderes als Befehlserweiterungen sind.

M&T-Forth 64 arbeitet wahlweise mit Kassette und Diskette zusammen. Das Prinzip des virtuellen Speichers, das heißt, daß sich alle »Screens« (Textabschnitte) auf der Diskette befinden, wurde hier aus Geschwindigkeitsgründen nicht angewandt.

24-KByte-RAM-Floppy

M&T-Forth 64 simuliert vielmehr eine RAM-Disk, die 40 Screens von jeweils 16 Zeilen à 38 Zeichen enthält. Diese RAM-Disk kann jederzeit auch teilweise auf Kassette oder der echten Diskette gesichert werden. Ebenso sind schon compilierte Spracherweiterungen lad- und speicherbar. Bei Diskettenbetrieb lassen sich auch Directories listen oder der Fehlerkanal lesen und schreiben. Da außerdem Befehle wie OPEN und CLOSE im Sprachkern eingebaut wurden, kann auch eine Druckerausgabe recht einfach erfolgen.

Alles was nicht in den Sprachkern von Forth gehört, wurde konsequent auf die Programmdiskette ausgelagert. Dazu gehören der Forth-Assembler, der Zeileneditor sowie Grafik- und Sound-Befehlserweiterungen. Diese werden ja nicht ständig benötigt, und würden sonst immer nur Platz im Computer wegnehmen. Das Nachladen des Editors mag am Anfang etwas umständlich wirken, hat dann aber den Vorteil, daß er vom Benutzer jederzeit erweitert oder völlig umgebaut werden kann, ohne den Sprachkern verunstalten zu müssen.

Daß Forth recht maschinennah und somit schnell ist, ist allgemein bekannt. Trotzdem enthalten viele gute Forth-Systeme Schnittstellen zur Maschinensprache und sogar einen Assembler. Ebenso verhält es sich bei M&T-Forth 64. Der mitgelieferte Assembler hat den Vorteil, daß in Assembler strukturiert programmiert werden kann. Zeitkritische Aufgaben können somit elegant und übersichtlich realisiert werden. Zusätzlich zum Assembler wird auch ein Forth-Monitor mitgeliefert. Dieser Mini-Monitor kann es zwar nicht mit dem SMON aufnehmen, leistet aber sehr gute Dienste, wenn man unter Forth schnell mal eine Möglichkeit braucht, »direkt an die Maschine« zu gehen.

Integrierter Assembler

Der mitgelieferte Editor ist dagegen recht spartanisch. Es ist ein einfacher Standard-Zeileneditor, wie ihn fast jede Forth-Version enthält. Ein komfortabler Full-Screen-Editor wäre zwar wünschenswert, doch läßt sich mit dem Zeileneditor auch recht gut arbeiten, insbesondere, da Forth ja keine Sprache ist, bei der man drauflosprogrammieren kann.

Sehr interessant sind die mitgelieferten Grafik- und Sound-Befehle für Forth. Da ist zum Beispiel ein einfacher Spriteeditor, (in Forth gerade 11 Zeilen lang) vorhanden.

Der Grundbefehlssatz wurde absichtlich klein gehalten, damit der Endbenutzer sich die zusätzlichen Befehle an seine individuelle Anwendung anpassen kann, ein in Forth ja ganz natürlicher Vorgang.

Eine Reihe von mitgelieferten zusätzlichen Beispielprogrammen, darunter das Nicht-Standard-Wort CASES, das trotzdem gerne verwendet wird, runden das Gesamtbild des M&T-Forth 64 ab.

Fazit

Insgesamt gesehen liegt mit M&T-Forth 64 eine Forth-Version vor, die zwar nicht durch enorme Befehlsvielfalt glänzt, wie andere Versionen dieser Sprache, dafür aber eine umso solidere Basis für Eigenentwicklungen ist. Da der FIG-Standard strikt eingehalten wird, dürfte die Anpassung von schon geschriebenen Forth-Programmen überhaupt keine Probleme aufwerfen. Alles in allem also eine Forth-Version, mit der Anfänger wie Fortgeschrittene gleichermaßen gut bedient sind.

(B. Schneider/ev)

Info: Happy Software, Markt & Technik Verlag AG, Hans-Pinsel-Str. 2, 8013 Haar bei München. Preis 98,- Mark.

Assembler ist keine Alchimie — Teil 10

Auf zum Endspurt: Die letzten 4 Assembler-Befehle werden in Angriff genommen (jedoch nicht die letzte Folge des Kurses). Wir steigen ein in ein unerhört interessantes, wenn auch nicht einfaches Thema: Die Interrupt-Technik.

Die Assembler-Befehle haben wir bis auf vier noch offenstehende alle behandelt. Diese vier, die alle mit dem Interrupt-Handling zusammenhängen, sollen diesmal unser Thema sein. Um nach längerer Zeit mal wieder auf den Titel dieser Serie zurückzukommen (nämlich die Alchimie!): Wenn wir diese 4 Befehle beherrschen, haben wir den ersten Schritt zum Meister der Assembler-Alchimie getan. Diese vier kleinen 1-Byte-Befehle öffnen uns eine geheime Pforte zu einem Universum an Programmier-Möglichkeiten, von dem wir bisher kaum zu träumen vermochten. Genug der Schwärmerei, erst kommt noch eine Menge Arbeit, die uns wohl mehrere Folgen dieser Serie in Atem halten wird.

Zuvor noch eine Bemerkung: es gibt kaum ein Thema im Rahmen der Programmierung in Assembler, welches so penetrant häufig Abstürze provoziert, wie das nunmehr angesteuerte! Falls Sie noch keine RESET-Taste an ihrem Computer haben, wird es nun höchste Zeit. Diese nützlichen Dinger werden inzwischen schon so preiswert angeboten (sehen Sie mal in den Kleinanzeigenteil!), daß Sie zur Grundausstattung eines Assembler Alchimisten zählen.

Was sind Interrupts?

Unser Computer ist — solange er eingeschaltet ist — ständig mit irgendwelchen Tätigkeiten beschäftigt. Im Direktmodus hängt er beispielsweise meistens in einer Warteschleife und harrt der Eingaben, im Programm-Modus arbeitet er sich mit Hilfe der Interpreterschleife durch einen Basic-Befehlstext hindurch und so weiter. Nun werden Sie ja sicher schon festgestellt haben, daß er im Direktmodus auch den Cursor blinken läßt, in beiden Modi die TI$-Uhr weiterzählt und weitere Dinge macht, die anscheinend so nebenher passieren. Schon in der ersten Folge dieser Serie aber haben wir einen Unterschied zwischen Mensch und Computer festgehalten: Der Mensch kann mehrere Dinge gleichzeitig tun, der Mikroprozessor ist nur fähig zu einer Arbeit pro Zeiteinheit. Weil aber diese Zeiteinheiten so unfaßbar kurz sind (etwa eine Millionstel Sekunde), haben wir Benutzer den Eindruck der Gleichzeitigkeit.

Wenn dem aber so ist, wie macht es der Computer, daß er beispielsweise ein Programm abarbeitet und trotzdem die TI$-Uhr weiterzählt? Durch Unterbrechungen (interrupt = unterbrechen) der gerade ausgeübten Tätigkeit. Ein Beispiel aus dem täglichen Leben soll uns das illustrieren: Sie lesen gerade diesen Artikel, als das Telefon klingelt und ein Freund von Ihnen wissen möchte, was eigentlich Unterbrechungen sind. Während Sie es ihm erklären, fängt in der Küche der Teekessel schrill zu pfeifen an. Sie sagen Ihrem Freund, er möge sich einen Moment gedulden, gehen in die Küche und nehmen den Kessel vom Feuer. Dann kehren Sie ans Telefon zurück und beenden nach einer Weile das Gespräch. Nach dem Auflegen des Telefonhörers setzen Sie die Lektüre des Artikels fort, fest entschlossen, sich nun nicht mehr unterbrechen zu lassen. Kurze Zeit später klingelt jemand an der Tür. Sie lassen sich dadurch nicht stören.

Dieses Gleichnis gibt ziemlich genau wieder, was sich im Computer — nur bei millionenfacher Geschwindigkeit — bei Unterbrechungen abspielt. In Bild 1 ist das Schema des Ablaufes grafisch dargestellt. In gewisser Weise ähnelt das ganze dem Abarbeiten von Unterprogrammsequenzen. Weshalb programmiert man dann nicht einfach mittels einiger JSR-Aufrufe? Dafür hat L.A.Leventhal einen einleuchtenden Vergleich: »Ein Unterbrechungs-System entspricht etwa einer Telefonklingel. Sie läutet, wenn ein Anruf empfangen wird, so daß man den Hörer nicht laufend abnehmen muß, um festzustellen, ob sich jemand in der Leitung befindet.« (L.A.Leventhal, »6502 Programmieren in Assembler«, München te-wi Verlag, S.121). Unterbrechungen können dann angefordert und abgearbeitet werden, wenn sie nötig sind, im Gegensatz zu Unterprogrammen, die erst dann berücksichtigt werden, wenn der Programmzähler einen JSR-Befehl erfaßt. Um also schnell reagieren zu können, müßte man sehr oft in einem Programm eine Unterroutine anspringen, die auf gewisse Registerinhalte prüft und dann zur Bearbeitung verzweigt oder — bei Nichtvorliegen einer Bedingung — im normalen Programm weiterfährt. Das kostet unnötig Zeit und Speicherraum. Mancher Verkehr des Computers mit Peripherie erfordert so schnelle Reaktionen, daß diese nur geleistet werden können durch Unterbrechen des laufenden Programmes.

Ich denke, daß Sie nun die Notwendigkeit von Unterbrechungen erkennen. Fast jede CPU kennt solche Unterbrechungssysteme. Man kann sie charakterisieren durch die Beantwortung folgender Fragen:

1) Welche Unterbrechungs-Eingänge weist die CPU auf?
2) Wie reagiert die CPU auf eine Unterbrechung?
3) Wie bestimmt die CPU die Unterbrechungsquelle, wenn die Anzahl der Quellen größer ist als die Anzahl der Eingänge?
4) Kann die CPU zwischen wichtigen und weniger wichtigen Unterbrechungen unterscheiden?
5) Wie und wann wird das Unterbrechungssystem freigegeben oder gesperrt?

All diese Fragen werden wir im Laufe dieser Serie für unseren Computer ergründen.

Das Unterbrechungssystem der CPU 6510/6502

Einige dieser Charakteristika sind schnell zu zeigen:

Zu 1: Unsere CPU hat ganau 2 Eingänge für Unterbrechungen (wenn man RESET außer acht läßt, was wir im folgenden meist tun werden).

Zu 3: Natürlich gibt es weitaus mehr denkbare Unterbrechungsquellen als diese 2 Eingänge, weshalb softwaremäßig eine Registerabfrage (das sogenannte Polling) durchgeführt wird, um die Quelle festzustellen.

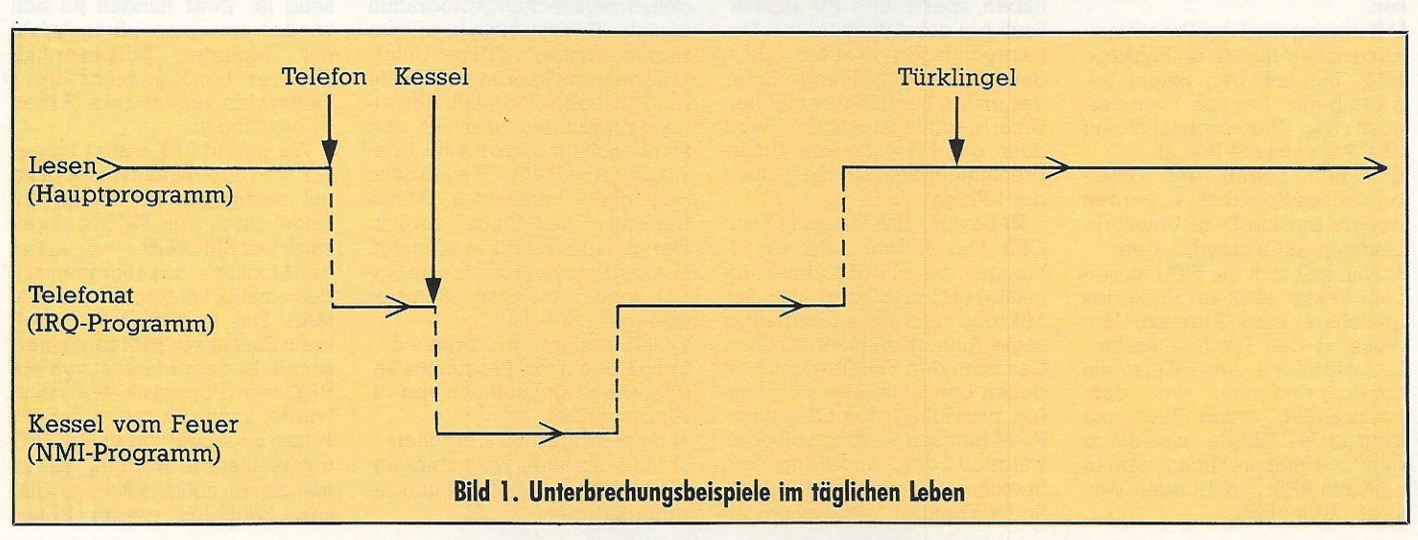


Bild 1. Unterbrechungsbeispiele im täglichen Leben

Zu 4: Zwischen wichtiger und nicht so wichtiger Unterbrechung kann unsere CPU unterscheiden durch die Priorität der beiden Eingänge. Wir haben eine sogenannte maskierbare Unterbrechung, genannt IRQ, welche per Befehl ignoriert (maskiert) werden kann und eine andere, nicht maskierbare, die daher auch NMI (not maskable interrupt = nicht maskierbare Unterbrechung) genannt wird. NMI hat eine höhere Priorität als IRQ und kann deshalb für die wichtigeren Aufgabenstellungen eingesetzt werden.

Zu 5: Freigegeben oder gesperrt werden kann die IRQ-Unterbrechung durch ein Sperrbit (auch Maskenbit genannt), welches sich als Bit 2 im Flaggen-Register des Prozessors befindet. Das ist die I-Flagge. Für den Empfang der NMI-Unterbrechung kann die CPU nicht gesperrt werden.

Um mal die Parallele zu unserem Beispiel zu zeigen: Das Lesen des Artikels ist die gerade stattfindende Tätigkeit des Computers. Die Telefonklingel signalisiert einen IRQ, der im folgenden bearbeitet wird. Das Pfeifen des Teekessels soll einem NMI entsprechen. Wenn dieser dann bearbeitet ist, geht es mit der Abarbeitung des IRQ weiter. Nach Beendigung des Telefonates wird das Unterbrechungs-Sperrbit gesetzt (sie nehmen sich vor, sich nicht mehr stören zu lassen) und mit der normalen Tätigkeit fortgefahren. Weil der nun folgende IRQ damit maskiert ist, wird das Türklingeln ignoriert.

Die **Frage 2**, nämlich wie unsere CPU auf eine Unterbrechung reagiert, blieb noch unbeantwortet. Nun soll sie behandelt werden:

a) Am Ende jedes Befehls überprüft die CPU automatisch den Zustand des Unterbrechungs-Systems. Wenn an einer der beiden Unterbrechungsleitungen eine Anforderung vorliegt und diese auch freigegeben ist, beginnt die Unterbrechung zu wirken.

b) Zunächst wird der Programmzählerinhalt in der Reihenfolge MSB, LSB auf den Stapel geschrieben. Danach wandert noch der Prozessorstatus auf den Stapel (siehe Bild 2).

c) Durch Setzen des Unterbrechungs-Sperrbits I werden weitere maskierbare Unterbrechungen (IRQ) unterbunden.

d) Nun holt sich die CPU aus einem Vektor ganz am Ende des Speichers eine Adresse, lädt diese in den Programmzähler und startet auf diese Weise ein Serviceprogramm, das dem auslösenden Anlaß Rechnung trägt. In der Tabelle 1 sind die zu den Unterbrechungsformen und zum RESET gehörigen Vektoren aufgeführt.

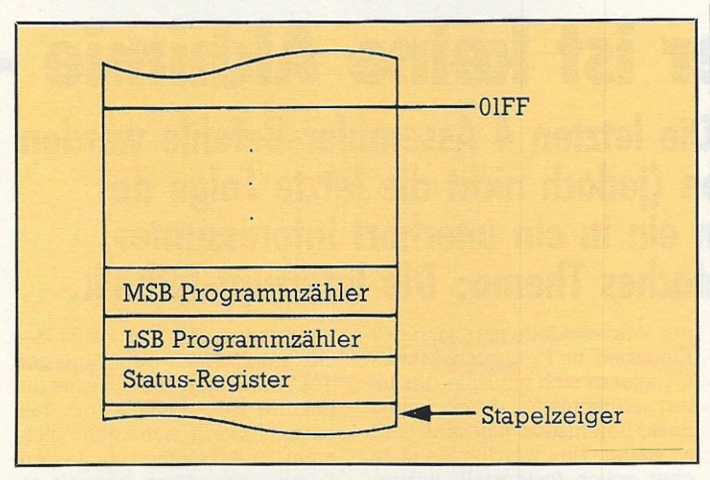


Bild 2. Die CPU rettet den Programmzähler und das Statusregister beim Eintreten einer Unterbrechung auf den Stapel

Bevor wir uns weiter mit den so angesteuerten Routinen befassen, wollen wir die 4 Befehle kennenlernen, die uns noch fehlen.

Schlüssel zur Unterbrechungsprogrammierung: CLI, SEI, RTI, BRK

Das Sperren der maskierbaren Unterbrechung IRQ und das Löschen der Maske erfolgt durch Setzen oder Löschen des Sperrbits im Prozessorstatus-Register. Dieses Bit, die I-Flagge, kann durch den Befehl CLI gelöscht werden. **CLI** kommt von »**CL**ear **I**nterrupt mask«, was bedeutet »lösche die Unterbrechungs-Maske«. Immer dann, wenn IRQs zugelassen sein sollen zur Bearbeitung durch den Mikroprozessor, muß damit die I-Flagge gelöscht werden. Wie Sie sehen, ist CLI ein 1-Byte-Befehl mit impliziter Adressierung. Er braucht genau 2 Taktzyklen zur Erledigung seiner Aufgabe.

Wenn wir später eigene Unterbrechungsroutinen schreiben, stehen wir oft vor der Frage, ob wir innerhalb unseres Unterbrechungsprogramms weitere Unterbrechungen zulassen wollen. Manchmal ist das wichtig, beispielsweise bei der Tastaturabfrage. Wie wir vorhin erwähnt haben, sperrt die CPU automatisch bei der Annahme von Unterbrechungen weitere IRQs durch Setzen der I-Flagge. Einer der ersten Befehle der eigenen Unterbrechungsroutine wird dann die Freigabe von Unterbrechungen sein durch Löschen der I-Flagge.

SEI bewirkt das Gegenteil von CLI. Der Befehl setzt die I-Flagge auf 1 (»**SE**t **I**nterrupt mask«) und verhindert, daß der Mikroprozessor weiteren IRQs seine Aufmerksamkeit schenkt. Das ist in den Fällen wichtig, in denen beispielsweise störungsfrei der Inhalt des Charakter-ROM gelesen werden soll oder während der Änderung von Speicherstellen, die die IRQ-Routine benutzt. Wie wichtig das Sperren von IRQs sein kann, haben Sie eventuell bemerkt, wenn Ihnen das Hilfsbildschirmprogramm aus der 6. Folge mal abgestürzt war. Seit der letzten Folge — wo wir die IRQs gesperrt haben — ist Ihnen das sicherlich nicht mehr passiert. Ebenso wie CLI ist SEI ein 1-Byte-Befehl mit impliziter Adressierung, und auch er braucht 2 Taktzyklen zur Bearbeitung.

Noch eine Bemerkung zum Verhindern der IRQs. Wir werden später sehen, was alles während der 60mal pro Sekunde aufgerufenen Unterbrechung erledigt wird. Jede Routine, die SEI verwendet, verbraucht Rechenzeit. Wenn sie so lange dauert, daß eine oder mehrere dieser regelmäßigen IRQs unterbunden werden, kann das unter Umständen zu Störungen von Programmabläufen führen. In solchen Fällen ist es sinnvoll, in die eigene Routine den Teil der regulären IRQ-Routine einzubauen, der im Programmablauf durch sein Fehlen Störungen verursacht. Meistens kann man aber durch gute Planung eines Programmes dieses Problem umgehen.

RTI heißt »**R**e**T**urn from **I**nterrupt«, zu deutsch also: »kehre aus dem Unterbrechungsprogramm zurück.« Es entspricht in seinem Einsatz etwa dem RTS bei Unterprogrammrücksprüngen. Während RTS aber lediglich den alten Programmzählerinhalt vom Stapel holt (und noch eine 1 dazuaddiert), schafft RTI auch noch den alten Inhalt des Status-Registers vom Stapel zurück. Der genaue Ablauf ist wie folgt:

1) Alten Prozessorstatus vom Stapel wieder ins Status-Register schieben.

2) Stapelzeiger um 1 erhöhen

3) LSB des alten Programmzählers vom Stapel nehmen und zurückschreiben.

4) Stapelzeiger um 1 erhöhen

5) MSB des alten Programmzählers vom Stapel nehmen und zurückschreiben.

6) Stapelzeiger um 1 erhöhen.

Damit ist der Zustand vor der Unterbrechung wieder hergestellt. Auch die I-Flagge ist so automatisch wieder gelöscht, denn vor der Unterbrechung war sie sicher nicht gesetzt gewesen und der alte Status-Zustand ist ja jetzt wieder vorhanden.

RTI ist ebenfalls ein 1-Byte-Befehl mit impliziter Adressierung. Seine vollständige Bearbeitung dauert 6 Taktzyklen.

Bei eigenen Unterbrechungs-Routinen verwendet man häufig nicht RTI, sondern springt durch JMP an eine sinnvolle Stelle des normalen Unterbrechungsprogrammes. Auf diese Weise kann man dann die normalen Arbeitsgänge der vorprogrammierten Unterbrechung oder Teile davon noch ausführen lassen.

Den Befehl **BRK** (break = Software-Unterbrechung) haben wir schon verwendet. Er entspricht in seinem Einsatz etwa dem STOP-Befehl in Basic und dient wie jeder Befehl dort hauptsächlich dem Testen von Programmen. Tatsächlich unterscheidet sich die Reaktion unserer CPU bei Auftreten eines BRK kaum von der bei einem IRQ. Folgendes passiert:

a) Der Programmzähler wird um 2 erhöht.

b) Bit 4 des Prozessorstatusregisters, die Break-Flagge B, wird auf 1 gesetzt.

c) Das MSB des Programmzählers wird auf den Stapel gebracht und der Stapelzähler um 1 heruntergezählt.

d) Dasselbe geschieht nun mit dem LSB des Programmzählers

e) und mit dem Statusregister.

f) Das Unterbrechungsmaskenbit, die I-Flagge, wird auf 1 gesetzt um IRQs zu sperren.

g) In den Programmzähler wird nun aus dem Vektor FFFE/FFFF dieselbe Adresse geladen, die auch bei IRQs benutzt wird. Damit startet nun das Programm, das diese Unterbrechung bearbeitet.

Sie sehen, daß der BRK-Befehl ein ziemlich komplizierter Geselle ist. Zwar handelt es sich wieder um einen 1-Byte-Befehl mit impliziter Adressierung, aber er benötigt immerhin 7 Taktzyklen, um all diese Arbeit zu bewältigen.

Wir haben BRK bisher immer zur Programmunterbrechung mit nachfolgender Registeranzeige durch den SMON eingesetzt. Der SMON ist — wie fast jeder Monitor — so programmiert, daß ein BRK zur Registeranzeige führt. Das ist natürlich sinnvoll beim Einsatz von BRK zur Fehlersuche. In dem Moment, wo ein BRK vom Prozessor bearbeitet wurde, kann nur durch die gesetzte B-Flagge von einem IRQ unterschieden werden. Es ist manchmal nötig, schon zu diesem Zeitpunkt diesen Unter-

schied festzustellen. Deshalb verwendet man den nachfolgend beschriebenen Test zu diesem Zweck:

PLA
in den Akku wird das zuletzt auf den Stapel geschobene Prozessorstatus-Register geholt.

PHA
und sogleich wieder zurückgeschoben

AND #$10
durch die AND-Verknüpfung mit der Binärzahl 0001 0000 kann eine eventuell vorhandene B-Flagge isoliert werden.

BNE BREAK
Falls eine B-Flagge gesetzt war, ist der Akku ungleich 0 und die Bearbeitung verzweigt zum von uns konstruierten BREAK-Programm. War der Akku nach dieser AND-Verknüpfung gleich 0, dann erfolgt keine Verzweigung und es handelt sich um einen IRQ, zu dessen Bearbeitung nun zu springen ist.

Es gibt noch eine andere — gebräuchlichere — Möglichkeit, zwischen einem BRK und einem IRQ zu unterscheiden, die allerdings erst zu einem späteren Zeitpunkt des computerinternen Unterprogrammes erfolgt. Von um 2 erhöht worden ist. Manchmal sind deshalb noch Korrekturen des Programms nötig.

Ich hoffe, daß Sie bisher diesen Artikel nicht zu frustrierend fanden, denn ständig ist die Rede vom eigenen Unterbrechungs-Programm und dabei wissen Sie — außer durch BRK — noch gar keine Möglichkeit, einen IRQ oder NMI auszulösen, und Sie sind sicher noch sehr vorsichtig mit dem Gedanken an eigene Unterbrechungs-Routinen, weil Ihnen ja noch unbekannt ist, wie die normale Firmware Unterbrechungen behandelt. Keine Angst: All das werden wir noch klären. Betrachten Sie diese Folge zum Thema Unterbrechungen vielleicht mehr wie ein Handbuch, in dem Sie dann, wenn Ihr Verständnis gestiegen ist, nochmal zurückblättern können:

Wir haben bisher nur betrachtet, wie unsere CPU reagiert, wenn an einem der beiden Unterbrechungs-Eingänge (IRQ und NMI) eine Unterbrechungs-Anforderung vorliegt. Um nun aber selbst ins Geschehen eingreifen zu können, ist es nötig zu wissen, wie diese Anforderung ne gewisse Übersicht zu bekommen, sollte man unterscheiden zwischen primären und sekundären Unterbrechungsquellen. Das soll kurz erläutert werden: Die Diskettenstation beispielsweise ist über den seriellen Port mit dem Computer verbunden. Dieser wiederum steht in direktem Kontakt zu 2 Bausteinen, den CIAs. Erst diese CIAs stehen in direktem Kontakt zur CPU. Alle Unterbrechungs-Quellen, die direkt Signale an die beiden Unterbrechungseingänge unserer CPU senden, sollen künftig »primäre« Quellen genannt werden, die anderen, die nur über solch eine primäre Quelle Unterbrechungs-Anforderungen stellen, werden von uns als »sekundäre« Quellen bezeichnet. Weil wir irgendwo einen Schnitt machen müssen — einmal, um nicht völlig auszuufern in der Erklärung von peripheren Geräten (das soll anderen, kompetenteren überlassen bleiben) und zum anderen, weil ich mich da auch nicht so gut auskenne — werden wir uns im folgenden auf die primären Unterbrechungsquellen beschränken. Da bleibt aber noch mehr als genug zu tun

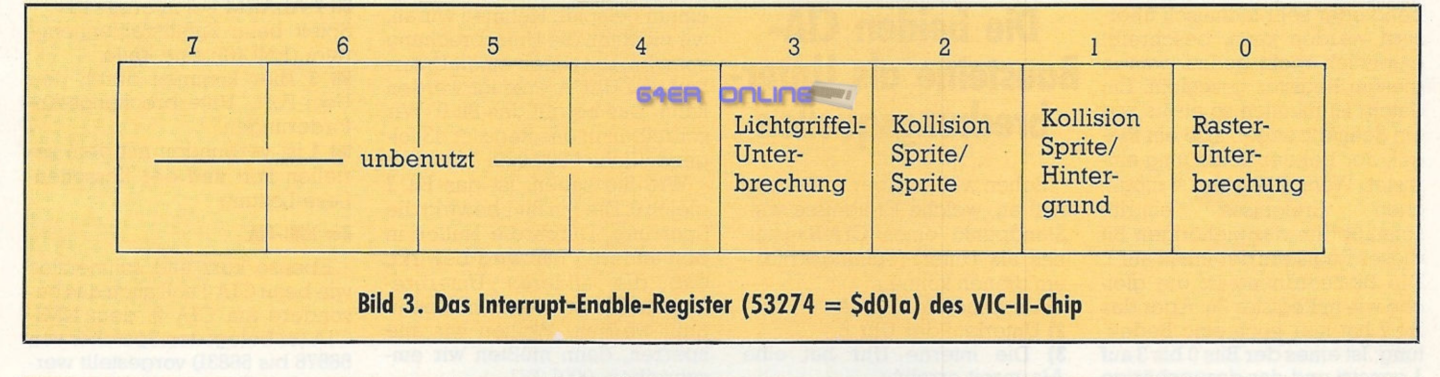
Den Expansion-Port werden wir nicht behandeln und einen RESET nur ziemlich kurz betrachten, weil es sich dabei eigentlich nicht um eine Unterbrechung im bisher definierten Sinn handelt.

Der VIC-II-Chip als Unterbrechungsquelle

Soweit ich feststellen konnte, kommt der VIC-II-Chip in Bezug auf unsere CPU nur als Anforderer von maskierten Unterbrechungen (IRQ) in Frage. Die Handhabung seiner Unterbrechungs-Anforderungen geschieht im VIC-II-Chip durch zwei Register. Vier Ereignisse sind eingeplant, deren Eintreten zur Unterbrechung führen kann:

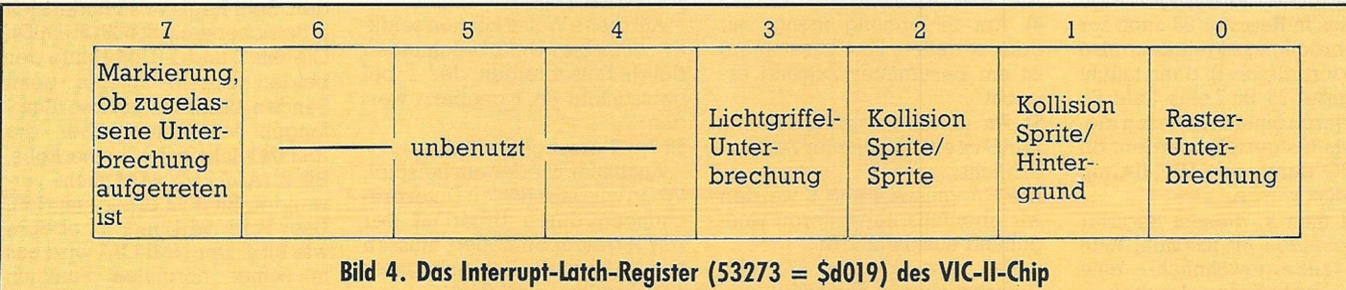
1) Rasterzeilen-Unterbrechung
2) Kollision eines Sprite mit Hintergrund
3) Kollision von Sprites untereinander
4) Lichtgriffel-Unterbrechung.

Die ersten 3 Auslöser werden wir uns in kommenden Folgen genau ansehen und dabei vielerlei interessante Möglichkeiten feststellen. Die Option, die der Lichtgriffel bietet, wird nicht behandelt werden: Meine Kenntnisse auf diesem Sektor sind nur gering (nobody is perfect).



7	6	5	4	3	2	1	0
—	— unbenutzt	—	—	Lichtgriffel-Unterbrechung	Kollision Sprite/Sprite	Kollision Sprite/Hintergrund	Raster-Unterbrechung

Bild 3. Das Interrupt-Enable-Register (53274 = $d01a) des VIC-II-Chip



7	6	5	4	3	2	1	0
Markierung, ob zugelassene Unterbrechung aufgetreten ist	—	unbenutzt	—	Lichtgriffel-Unterbrechung	Kollision Sprite/Sprite	Kollision Sprite/Hintergrund	Raster-Unterbrechung

Bild 4. Das Interrupt-Latch-Register (53273 = $d019) des VIC-II-Chip

dieser zweiten Möglichkeit wird im SMON Gebrauch gemacht und wir werden sie nachher auch kennenlernen.

Natürlich kann der BRK-Befehl auch zu anderen Zwecken als zur Registeranzeige durch einen Monitor verwendet werden. Es kommt immer darauf an, welches Service-Programm wir dem Computer anbieten. Springt man aus so einem Service-Programm mittels RTI zurück ins Hauptprogramm, dann muß man berücksichtigen, daß der Programmzähler vor der Sicherung auf dem Stapel

dorthin gelangt. Das erfordert von uns die Beschäftigung mit anderen Computerbausteinen als der CPU, die bisher im Mittelpunkt unseres Interesses stand.

Woher kommen die Unterbrechungs-Anforderungen?

Quellen für Unterbrechungen können viele genannt werden: Diskettenstation, Datasette, Drucker, Modem, Schaltelemente und so weiter. Um aber ei-

übrig und deshalb soll auch nur eine Auswahl dieser Primärquellen detailliert behandelt werden.

Welches sind nun die primären Unterbrechungsquellen? Hier sind sie aufgeführt:

1) Der VIC-II-Chip (MOS 6566/6567 Video Interface Controller)
2) Die beiden CIAs (MOS 6526 Complex Interface Adapter)
3) Die RESTORE-Taste
4) Der Expansion-Port
5) RESET (paßt hier nicht ganz her, woanders aber auch nicht besser)

Das sogenannte Interrupt Enable Register (Unterbrechungs-Zulassungs-Register) des VIC-II-Chips ist Register 26. Es befindet sich in der Speicherstelle 53274 ($D01A) (siehe Bild 3).

In diesem Register wird festgelegt, ob eines — oder mehrere — der 4 möglichen auslösenden Ereignisse eine Unterbrechungsanforderung an den Mi-

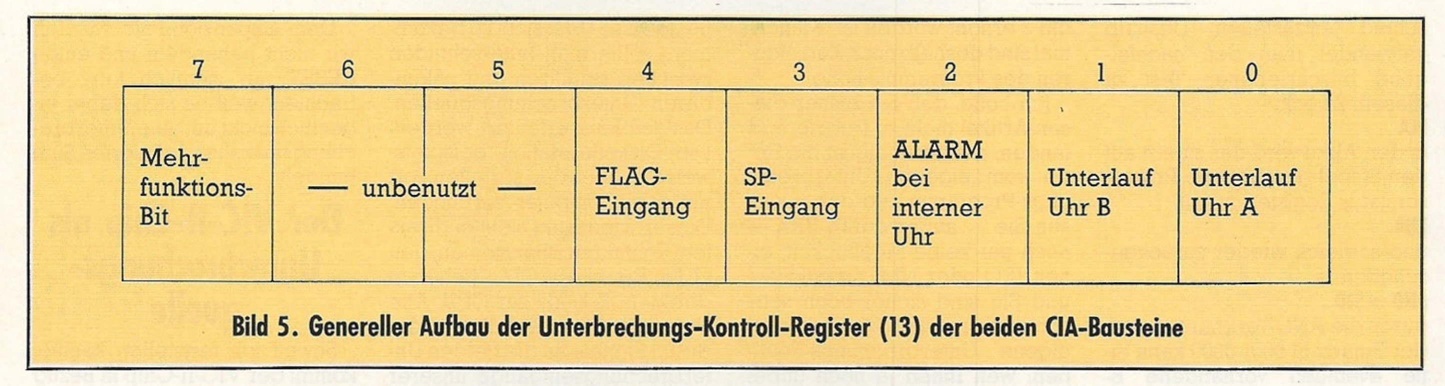


Bild 5. Genereller Aufbau der Unterbrechungs-Kontroll-Register (13) der beiden CIA-Bausteine

kroprozessor senden soll. Jedem Ereignis ist ein Bit zugeordnet. Ist dieses Bit gleich 1, dann ist die Unterbrechung freigegeben, ist es gleich 0, dann liegt eine Sperrung vor. Die Zuordnung der Bits ist wie folgt:

Bit 0 Rasterzeilen-IRQ
Bit 1 Sprite/Hintergrund-Kollision
Bit 2 Sprite/Sprite-Kollision
Bit 3 Lichtgriffel-IRQ
Bits 4 bis 7 sind ungenutzt und haben immer den Wert 1.

Das Register 25 wird Interrupt Latch Register genannt, was etwa zu übersetzen wäre mit »Unterbrechungs-Einrast-Register« (siehe Bild 4). Der englische Ausdruck »latch«, der nur umschreibend oder sehr technisch übersetzt werden kann, beschreibt eigentlich recht genau, was in diesem Register geschieht. Ein »latch« ist nämlich so etwas wie ein Schnappriegel, also ein Riegel, der bei der Betätigung einrastet. Wenn eines der 4 möglichen Ereignisse eintritt, schnappt im dazugehörigen Bit dieses Registers der Inhalt auf 1. Die Bit-Zuordnung ist die gleiche wie in Register 26. Aber das Bit 7 hat hier noch eine Bedeutung: Ist eines der Bits 0 bis 3 auf 1 gesetzt und das dazugehörige Ereignis in Register 26 auch zur Unterbrechung zugelassen (also auch dort gleich 1), dann taucht in Register 25, Bit 7 eine 1 auf. So kann durch einfaches Lesen dieses Bits festgestellt werden, ob ein IRQ durch den VIC-II-Chip ausgelöst wurde.

Will man in diesem Register ein gesetztes Bit löschen, muß man — außergewöhnlich! — eine 1 in die Bitposition schreiben.

Mit Recht erwarten Sie nun eigentlich eine Anwendung des bisher gelernten. Bei Unterbrechungsprogrammen ist es aber dringend nötig, immer den gesamten Komplex im Auge zu haben. Ich habe mich daher entschlossen, zuerst alles zu erklären und dann Anwendungsmöglichkeiten vorzustellen. Ihre Geduld wird auf eine harte Probe gestellt, aber ich hoffe, daß Sie ab der nächsten Folge feststellen, daß es sich gelohnt hat, etwas zu warten.

An sich sind die beiden CIAs in unserem Computer völlig identisch. Sie werden aber unterschiedlich eingesetzt. Sehen wir uns zunächst einmal an, was beiden in Bezug auf Unterbrechungen gemeinsam ist, um danach die Unterschiede festzuhalten. Die Unterbrechungs-Steuerung geschieht in Register 13 dieser Bausteine. Dieses Register hat 2 Funktionen: Es bestimmt, ob eine Unterbrechungsanforderung an die CPU gesandt werden soll, und es stellt fest, ob ein Ereignis stattgefunden hat, das zur Unterbrechung führen kann. Die Bedienung dieses Registers ist demzufolge auch etwas unübersichtlich, aber wir haben schon ganz andere Probleme gemeistert.

Die beiden CIA-Bausteine als Unterbrechungsquellen

Sehen wir uns aber zuerst einmal an, welche Ereignisse vom Standpunkt eines CIA-Bausteines als Unterbrechungskriterium dienen können:

1) Unterlauf der Uhr A
2) Unterlauf der Uhr B
3) Die interne Uhr hat eine Alarmzeit erreicht
4) Am SP-Eingang (hängt mit dem seriellen Port zusammen) ist ein bestimmter Zustand erreicht
5) An einem Eingang namens FLAG ist ein bestimmter Zustand erreicht.

Die Ereignisse 4 und 5 werden wir ebenfalls im weiteren weitgehend ausklammern.

Nun zum Register 13, dem Unterbrechungs-Kontroll-Register (siehe Bild 5).

Auch hier gehört zu jedem Ereignis ein Bit. Dabei — um Wiederholungen zu vermeiden — ist die Zuordnung schon durch die eben angegebene Ereignisaufzählung gegeben. Ziehen Sie von der vorangestellten Nummer immer eine 1 ab und Sie haben die Bitnummer. Die Bits 5 und 6 sind unbenutzt. Bit 7 hat eine dreifache Funktion, die eng mit den anderen Bitinhalten verknüpft ist. Sehen wir uns das mal der Reihe nach an:

Lesen des Registers

Sind Unterbrechungsereignisse aufgetreten, dann sind die dazugehörigen Bits auf 1 gesetzt. Bit 7 ist gleich 1, wenn mindestens ein solches Ereignis stattgefunden hat und außerdem dieses Ereignis als Unterbrechungsauslöser freigegeben ist. Auf diese Weise kann — ähnlich wie beim VIC-II-Chip-Register 25 — festgestellt werden, ob die Unterbrechung durch einen der beiden CIAs angefordert wurde. Im Unterschied aber zum VIC-II-Register wird Register 13 durch das Lesen **gelöscht**. Braucht man den Inhalt also noch, sollte man ihn irgendwo zwischenspeichern.

Schreiben in das Register

Bit 7 = 0 erzeugt Sperren.

Das erkennt man am besten an einem Beispiel. Nehmen wir an, wir möchten die Unterbrechung sperren, die durch einen Unterlauf von Uhr A erzeugt werden kann. Das betrifft das Bit 0. Wir schreiben in das Register 13 folgende Zahl: 0000 0001

Wie Sie sehen, ist das Bit 7 gleich 0. Die 1 in Bit 1 bewirkt die Sperrung. Durch die Nullen in den anderen Bits wird bewirkt, daß die anderen Unterbrechungs-Ereignisse nicht beeinflußt werden. Wollten wir alle sperren, dann müßten wir einschreiben: 0001 1111

Auf diese Weise können selektiv einzelne Unterbrechungen durch Einschreiben der 1 bei gelöschtem Bit 7 gesperrt werden.

Bit 7 = 1 erzeugt Freigabe.

Auch hier wieder ein Beispiel. Wenn wir ganz gezielt Unterbrechungen durch Unterlauf der Uhr A freigeben wollen, müssen wir die folgende Zahl in Register 13 schreiben: 1000 0001

Bit 7 (gleich 1) zeigt an, daß diejenigen Unterbrechungen freizugeben sind, deren Bits auf 1 gesetzt sind. Alle anderen Unterbrechungen, wo also in der dazugehörigen Bitposition der einzuschreibenden Zahl eine 0 steht, bleiben unverändert.

Ein wichtiger Unterschied zwischen den beiden CIAs ist der, daß der Unterbrechungsausgang von CIA 1 mit dem IRQ-Eingang der CPU verbunden ist, wohingegen der entsprechende Ausgang von CIA2 an den NMI-Eingang unseres Mikroprozessors führt. Daher löst der CIA 1 nur IRQs aus, er wird manchmal deshalb auch IRQ-CIA genannt. Der andere ist dann der NMI-CIA, weil er nur NMIs anfordern kann.

Der IRQ-CIA

Das Register 13 des IRQ-CIA (der die Speicherstellen 56320 bis 56335 belegt), liegt in Zelle 56333 ($DC0D). Die einzelnen Bits sind wie folgt zugeordnet:

Bit 0 Unterlauf Uhr A
Von hier kommt der IRQ, der 60mal pro Sekunde stattfindet zur Tastaturabfrage, zum Weiterstellen der TI$-Uhr etc.
Bit 1 Unterlauf Uhr B
Spielt bei Kassettenoperationen und dem seriellen Port eine Rolle.
Bit 2 ALARM bei interner Uhr.
Spielt beim Zufallszahlengenerator (RND(0)) eine Rolle.
Bit 3 Hier kommen durch den User-Port Unterbrechungs-Anforderungen.
Bit 4 ist verbunden mit dem seriellen Port und der Kassetten-Lese-Leitung.

Der NMI-CIA

Ebenso kurz und schmerzlos wie beim CIA 1 soll auch das besondere am CIA 2, dem NMI-CIA (er belegt den Speicher von 56576 bis 56831) vorgestellt werden. Sein Register 13 findet sich in Speicherstelle 56589 ($DD0D). Die Bits 0 und 1 (Unterläufe der beiden Uhren) spielen beim Senden beziehungsweise Empfangen von Daten über die RS232C-Schnittstelle eine Rolle, Bit 2 (ALARM) wird nicht verwendet, Bit 3 ist direkt mit dem User-Port verbunden ebenso wie Bit 4. Der NMI-CIA wird uns in seiner normalen Funktion nicht mehr beschäftigen.

Die RESTORE-Taste und ein kleines Testprogramm

Die RESTORE-Taste ist direkt mit dem NMI-Eingang unseres Mikroprozessors verbunden. Das ermöglicht es uns, durch einfaches Drücken dieser Taste jederzeit ins Geschehen einzugreifen, ohne uns um Details kümmern zu müssen, ob sich der Computer gerade im Direkt- oder im Programm-Modus befindet und so weiter. Denn NMI hat die höchste Priorität der Unterbrechungen.

Ein kleines Testprogramm soll Ihnen hier noch vorgestellt wer-

den, das Sie vielleicht aber noch nicht ganz verstehen werden, weil wir erst in der nächsten Folge die eingebauten Serviceprogramme kennenlernen werden. Schalten Sie also den SMON ein und geben Sie das Programm 1 ein (ab $6000):

6000	PHA	mit diesen Befehlen retten wir Akku und Register auf den Stapel.
6001	TXA	
6002	PHA	
6003	TYA	
6004	PHA	
6005	LDA #$7F	0111 1111 ist das in binär.
6007	STA $DD0D	Dadurch werden alle NMIs, die vom CIA 2 kommen könnten, gesperrt. Erinnern Sie sich: Bit 7 ist Null beim Schreiben, also Sperrfunktion.
600A	LDY $DD0D	Lesen des Registers 13 löscht dieses und zeigt uns, ob die NMI-Anforderung von dort kam.
600D	BMI $601A	falls NMI-Anforderung vom CIA 2 kam, wird verzweigt
600F	LDA $D020	ansonsten kommt der NMI von der RESTORE-Taste, und in den Akku wird die Rahmenfarbe eingeladen
6012	EOR #$0E	Ausgehend davon, daß als Rahmenfarbe 14 vorliegt, wird diese exklusiv oder verknüpft zu Null. Ist die Rahmenfarbe 0, dann wird sie wieder 14.
6014	STA $D020	Einschreiben des neuen Farbwertes
6017	JMP $FEBC	Sprung in den Rest der normalen NMI-Routine
601A	JMP $FE72	Sprung in die normale NMI-Routine im Fall, daß die Anforderung durch den NMI-CIA kam.

Programm 1. Ein kleines Testprogramm demonstriert die Wirkung einer Unterbrechung: Durch Drücken der RESTORE-Taste wird die Rahmenfarbe geändert.

Am besten speichern Sie nun das Programm ab und schalten dann mittels dem SMON-Kommando M 0318 die Anzeige der Bytes ab $0318 ein. Dort steht in den beiden ersten Speicherzellen 47 und FE. Mit dem Cursor fahren Sie in diese Zeile und ändern den Inhalt in 00 und 60, also unsere Programmstartadresse in der LSB/MSB-Form. Nach einem RETURN läuft nun jede NMI-Anforderung über unser Programm. Nun können Sie es ausprobieren, indem Sie mal die RESTORE-Taste drücken. Es genügt völlig, alleine diese Taste zu betätigen. Das wirkt — sichtbar durch die Änderung der Rahmenfarbe — in jedem Modus und jederzeit. Eine kleine Merkwürdigkeit ist, daß man manchmal etwas Geduld aufbringen muß, bis man die Wirkung sieht. Ich vermute, daß der NMI so schnell erledigt wird, daß sich mehrer NMIs pro Tastendruck ereignen. Man müßte sich noch eine kleine Routine überlegen, die die Wirkung etwas verzögert, denn 2 solche EOR-Kommandos nacheinander heben sich gegenseitig auf.

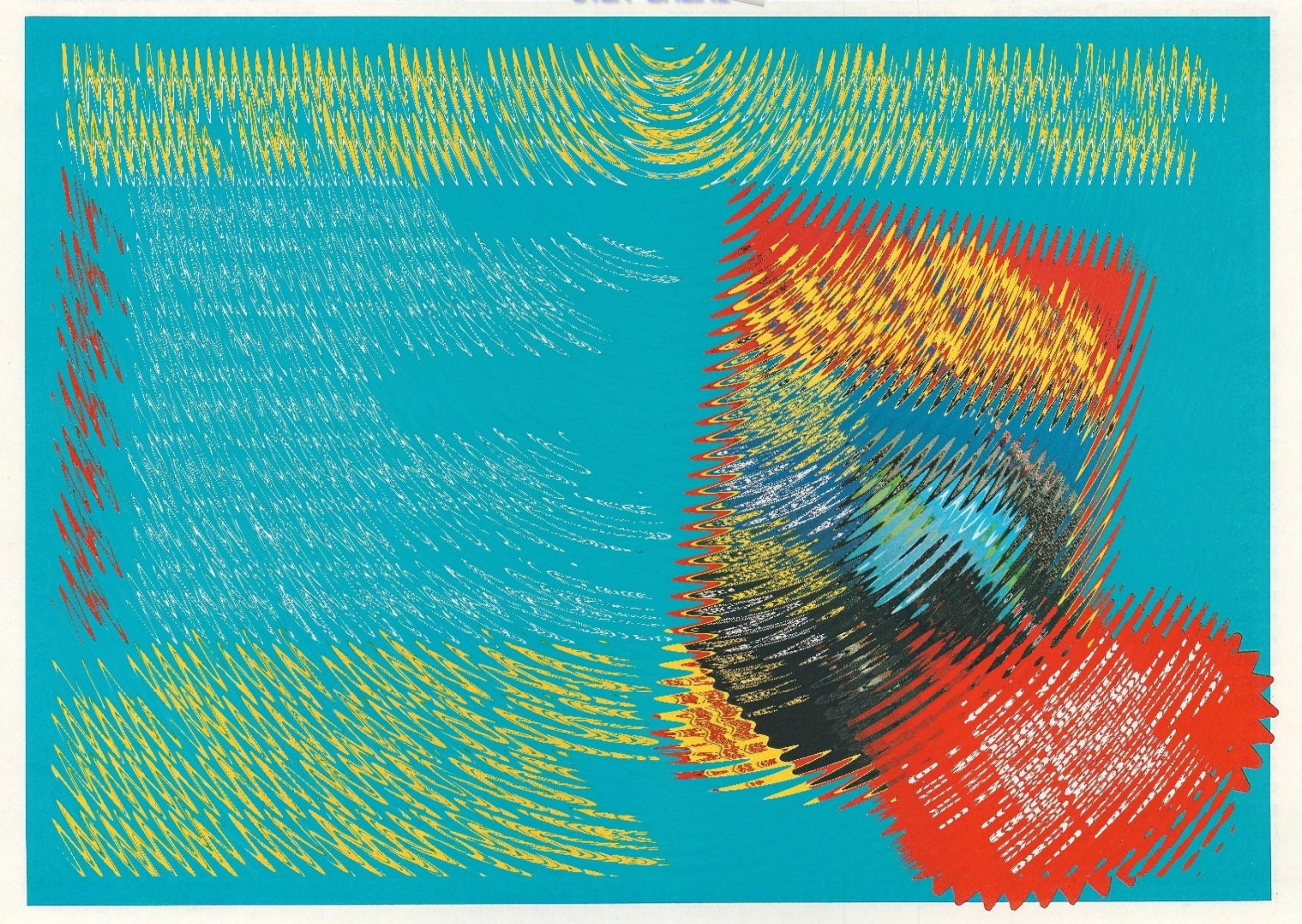
Damit sei's für diesmal genug. In der nächsten Ausgabe werden wir uns die Unterbrechungs-Firmware ansehen und einige Programmbeispiele vorstellen. Zum Schluß noch wie üblich eine Aufstellung (Tabelle 2) mit den besprochenen Befehlen. (Heimo Ponnath/gk)

Unterbrechungsart	Vektor	Zieladresse
Maskierbare Unterbrechung (IRQ, BRK)	$FFFE/FFFF	65352 $FF48
Reset	$FFFC/FFFD	64738 $FCE2
Nichtmaskierbare Unterbrechung (NMI)	$FFFA/FFFB	65091 $FE43

Tabelle 1. Unterbrechungsvektoren und ihre Inhalte

Befehls-wort	Adressie-rung	Byte-zahl	Code Hex	Code Dez	Takt-cyclen	Beeinflussung von Flaggen
CLI	implizit	1	58	88	2	I-Flagge
SEI	implizit	1	78	120	2	I-Flagge
RTI	implizit	1	40	64	6	alle Flaggen
BRK	implizit	1	00	0	7	B-Flagge vor dem Schieben auf den Stapel, I-Flagge danach

Tabelle 2. Die Daten zu den letzten Assembler-Befehlen



Dem Klang auf der Spur (Teil 7)

Die praktische Anwendung der bisher erarbeiteten Kenntnisse steht im Mittelpunkt dieser Folge: Ein universelles Synthesizer-Programm bringt den SID auf Trab.

Die meisten modernen digitalen Synthesizer haben eine Gemeinsamkeit: eine fast unüberschaubare Vielzahl von Einstellmöglichkeiten. Das in den letzten Folgen beschriebene Programm Modulator ist da keine Ausnahme. Um die reichhaltigen Möglichkeiten auszuloten, die dieses Programm bietet, benötigt man ein komfortables Editorprogramm, sonst läßt die Motivation beim Jonglieren mit Bits, Bytes, PEEK und POKE bald nach. Ein solches Programm wird hier vorgestellt. Es zeigt die über 70 Parameter des SID und von Modulator, übersichtlich nach Funktionsgruppen geordnet, auf dem Bildschirm an und erlaubt einen schnellen Zugriff auf alle Einstellwerte. Diese Einstellwerte können geändert werden, während ein Ton klingt, so daß die Wirkung der Einstellung unmittelbar überprüft werden kann. Zudem kann man Töne im Bereich von zwei Oktaven direkt über die Tastatur anspielen. Dabei können zwar nicht mehrere Tasten gleichzeitig gespielt werden, man kann aber zwei oder drei SID-Stimmen mit einem Tastendruck gleichzeitig erklingen lassen. Darüber hinaus verwaltet das Programm 24 Parametersätze, im folgenden Sounds genannt, im Arbeitsspeicher. Diese 24 Sounds können als eine Datei von Diskette geladen, bearbeitet und wieder abgespeichert werden.

Wie man den Sound-Editor zum Laufen bringt

Das abgedruckte Programm kann man beim Listing-Service bestellen oder auch selbst abtippen, was bei 107 Blöcken leider etwas mühsam ist. Dabei kann man die Kommentare natürlich weglassen, denn man hat sie im Heft ja bereits vor sich. Man sollte dann aber trotzdem alle Zeilennummern eintippen, da viele Zeilen angesprungen werden, auch wenn sie nur Kommentare enthalten. Reine Kommentarzeilen kann man mit: »(Zeilennummer) REM« oder »(Zeilennummer) : « abkürzen. Man hüte sich aber davor, eine RENUMBER-Utility auf das Programm anzuwenden. Die meisten Sprünge werden nämlich nicht über die Basic-Befehle GOTO und GOSUB abgewickelt, sondern über zwei Maschinenprogramme GT und GS (darüber später mehr). Diesen Umstand kann eine RENUMBER-Utility natürlich nicht berücksichtigen.

Der Sound-Editor ist nur zusammen mit dem in Ausgabe 4/85 (MSE-Listing) und in Ausgabe 5/84 (Assembler-Listing) veröffentlichten Programm Modulator lauffähig. Dieses Programm muß auf Diskette als absolut zu ladende Datei (Bereich $C075 bis $C422) unter dem Namen »MODULATOR« vorliegen.

Im Programm Modulator steckt leider noch ein kleiner Bug: Es wird der Zeropage-Speicherplatz $FF verwendet, was hin und wieder den Basic-Interpreter stören kann. Wenn man statt $FF den vom Betriebssystem nicht verwendeten Speicherplatz $02 verwendet, treten keine Probleme mehr auf. Im Modulator-Code sind zehn Zeropage-Adressen von dieser Änderung betroffen. Die folgenden vier Basic-Zeilen enthalten die zehn Adressen und nehmen auch gleich die Änderung vor:

```
10 FOR I=1 TO 10
20 READ AD:POKE AD,2:NEXT I
30 DATA 49339,49344,49734,
   49744,50115
40 DATA 50129,50162,50168,
   50173,50178
```

Der geänderte Modulator-Code sollte dann unter dem Namen »MODULATOR« auf Diskette zurückgespeichert werden. Jetzt müßte der Sound-Editor, sofern er fehlerfrei eingetippt wurde, laufen. Da es sich hier um ein Programm handelt, das überwiegend in Basic geschrieben ist, ist ein umfangreicher Setup-Teil, der etwa 30 Sekunden beansprucht, leider unvermeidbar. Das Warten wird dann allerdings durch Komfort und durchaus akzeptable Geschwindigkeit belohnt.

Mit 8 Menüs über 70 Parameter im Griff

Zum Verständnis der Parameter sei auf Ausgabe 4/85 verwiesen, wo die Funktionen der Modulatorkomponenten beschrieben sind. Der Sound-Editor meldet sich mit dem Hauptmenü (Bild 1) mit dem Namen MODULATOR. Damit soll betont werden, daß Modulator der eigentliche Kern des Programms ist. Der Sound-Editor wird ausschließlich über die Tastatur bedient, wobei sich die meisten Funktionen mit einem einzigen Tastendruck auslösen lassen. Es ist durchgehend das WYSIWYG-Konzept (What You See Is What You Get) verwirklicht, das heißt, alle in einer Situation verfügbaren Kommandos werden am Bildschirm angezeigt. Eine Ausnahme bilden die sieben Kommandos zur Auswahl der Untermenüs, die zwar nur im Hauptmenü angezeigt werden, aber auch in jedem Untermenü funktionieren. Man kann also direkt von einem Untermenü ins andere springen. Den Umweg über das Hauptmenü benötigt man eigentlich nur, wenn man die Kommandos nicht mehr weiß.

Das untere Drittel des Bildschirms ist in allen Menüs identisch. Es zeigt eine stilisierte Klaviatur über zwei Oktaven, der auf dem Computer die oberen beiden Tastenreihen entsprechen. Mit diesen Tasten kann man bis zu drei SID-Stimmen mit einem Tastendruck anspielen. Mit dem Kommando V und einer Ziffer 1, 2 oder 3 können die zu spielenden Stimmen aktiviert oder desaktiviert werden. Angezeigt wird das rechts neben der Klaviatur. Die Funktion SUSTAIN bewirkt, daß der Ton nach dem Loslassen einer Ton-Taste weiterklingt, sofern ein SUSTAIN-Pegel größer als Null eingestellt ist. Die GATE-Bits des SID werden bei aktivierter SUSTAIN-Funktion nicht zurückgesetzt. Mit dem Drücken der Space-Taste wird das GATE-Bit des Software-Hüllkurvengenerators (Soft-EG) gesetzt, durch Loslassen zurückgesetzt. Der Soft-EG wird also durch SPACE ausgelöst, vorausgesetzt, seine Parameter haben sinnvolle Werte. Das Auslösen des Soft-EG kann man mit Shift Space an die Tontasten koppeln, was bei den meisten Soft-EG-Anwendungen sinnvoll sein dürfte. Auch die SUSTAIN-Funktion und die Koppelung des Soft-EG werden rechts neben der Klaviatur angezeigt.

Die Beeinflussung der Parameter in den Untermenüs erfolgt nach einheitlichem Schema: Der zu ändernde Parameter wird mit den Cursor-Tasten angewählt und mit den f-Tasten verstellt. Je nach Anzahl der Parameter sind dabei alle vier Cursortasten oder nur die Tasten Cursor-Links und Cursor-Rechts wirksam. Auch die Funktionsweise der f-Tasten ist einheitlich: Numerische Parameter werden durch f1 und f7 in großer und durch f3 und f5 in kleiner Schrittweite verstellt. Bei nichtnumerischen Parametern wird die Funktion der f-Tasten am Bildschirm angezeigt.

Bei numerischen Parametern werden bis auf eine Ausnahme direkt die Werte der SID- oder Modulator-Register angezeigt. Bei der Ausnahme handelt es sich um die Basisfrequenzen der drei Stimmen (Bild 2, Parameter F). Bei dieser Basisfrequenz handelt es sich um die Frequenz des tiefsten spielbaren Tones (Taste Q). Sie wird nicht in Hertz, sondern in einer musikalisch sinnvollen Notation angegeben. Sie besteht aus Notennamen, einer Oktavnummer und einer Verstimmung in Cent. Die Oktavnummern bewegen sich im Bereich 0 bis 5. Dabei entspricht A 4 dem 440-Hz-Ton, der in der Musik korrekt »eingestrichenes a« heißt. Ein Cent ist ein hundertstel Halbton. Der Parameter bewegt sich von —49 bis +50. Die etwas komplizierte Berechnung der Basisfrequenz in Hz und in SID-Registerwerten findet man in den Programmzeilen ab 3110.

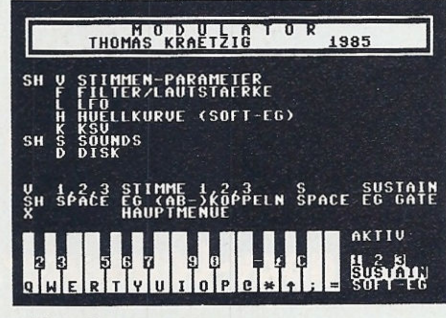
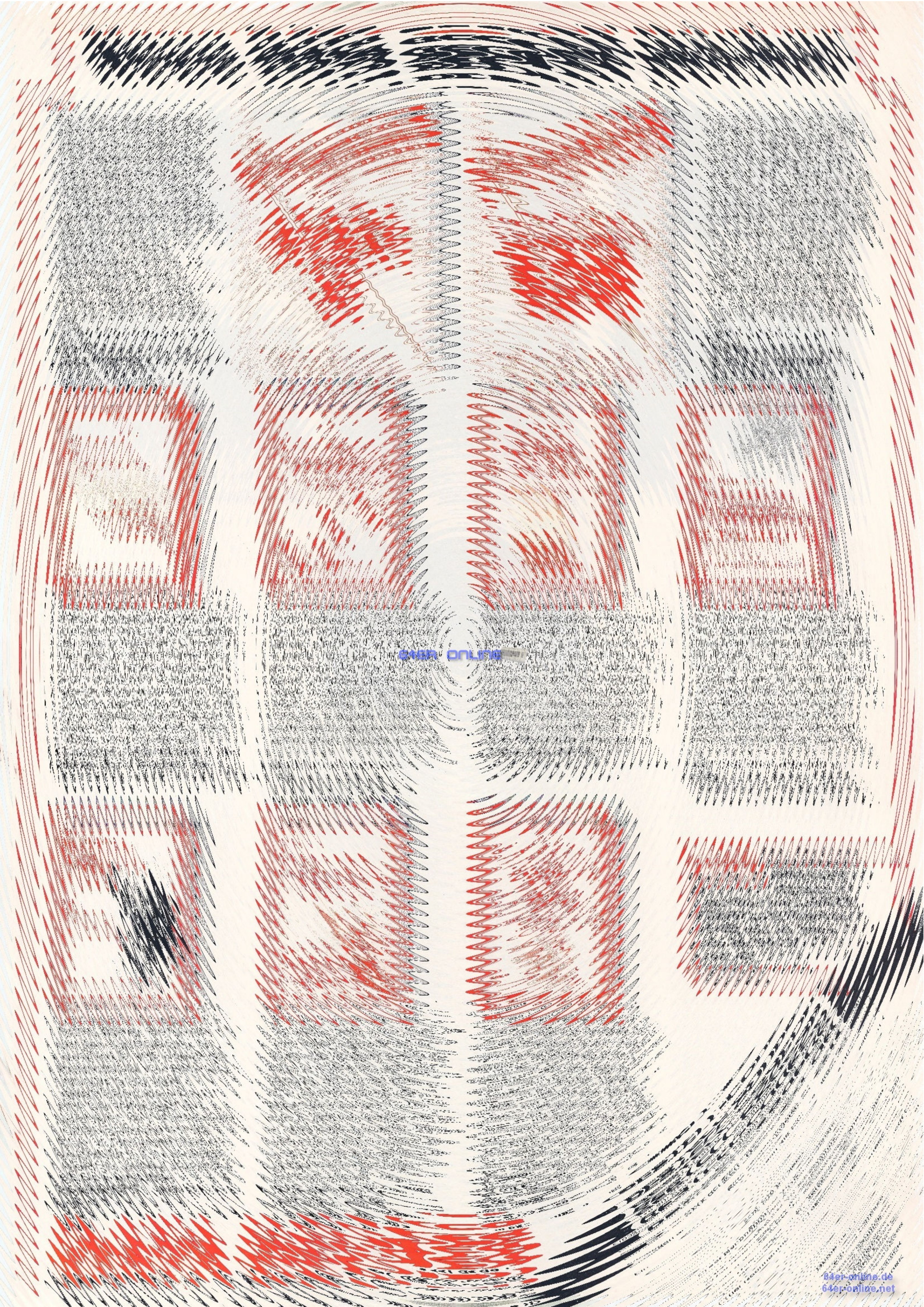


Bild 1. Hauptmenü

Alle anderen Sound-Parameter sind anhand der Menübilder, in denen sie auftreten (Bild 2 bis 6), leicht identifizierbar und bedürfen keiner weiteren Erläuterung.

Im Menü SOUNDS (Bild 7) können ganze Parametersätze (jeweils 107 Byte, siehe auch Tabelle 4) mit Namen versehen und im Speicher abgelegt werden. Angewählt


64er online

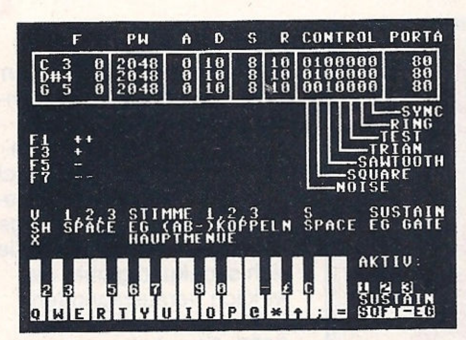


Bild 2. Untermenü Stimmen-Parameter

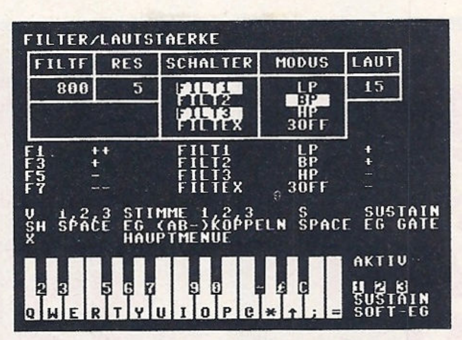


Bild 3. Untermenü Filter/Lautstärke

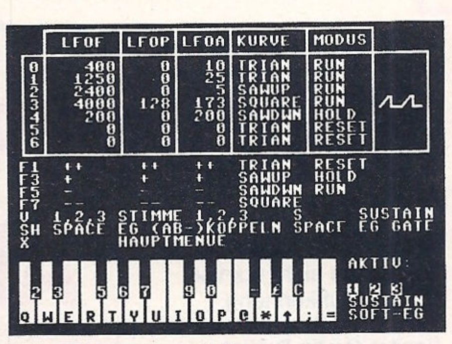


Bild 4. Untermenü LFO

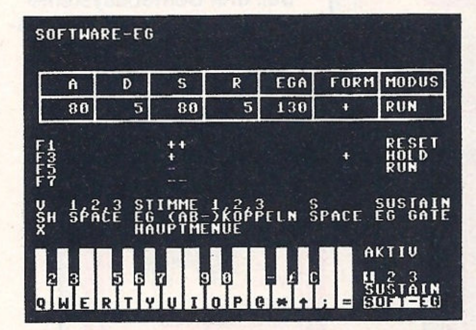


Bild 5. Untermenü Software-EG

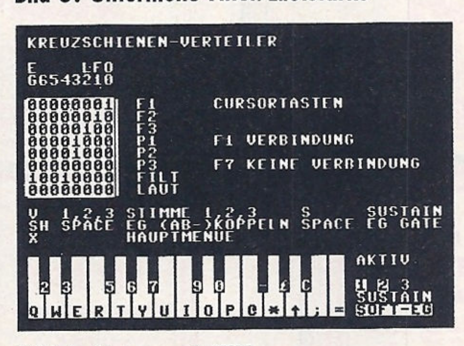


Bild 6. Untermenü KSV

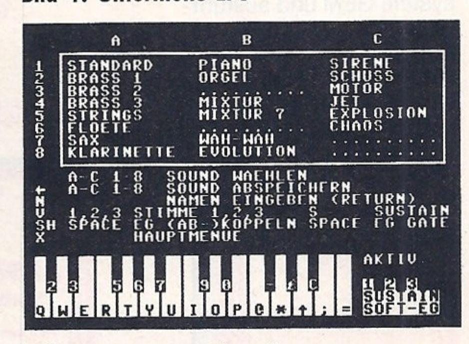


Bild 7. Untermenü Sounds



Bild 8. Untermenü Disk

werden die Sounds hier nicht über die Cursor-Tasten, sondern über die Tastenkombinationen A1 bis C8. Die 24 Sounds sind alle mit der Parametereinstellung, die beim Programmstart vorliegt, vorbesetzt. Eine Gruppe von 24 Sounds kann im Menü DISK (Bild 8) auf Diskette abgespeichert und wieder geladen werden.

Bemerkungen zum Programm

Es soll hier nicht die Arbeitsweise des Sound-Editors im Detail beschrieben, sondern nur einige Bemerkungen zum Stil des Programms gemacht werden. In dem Programm werden sehr viele Tastenkommandos akzeptiert, die in verschiedenen Menüs und bei den verschiedenen Parametern auch noch unterschiedliche Bedeutung haben. Um in dieser Situation die Übersicht zu behalten (Anliegen des Programmierers) und um das Programm trotz Realisierung in Basic akzeptabel schnell zu halten (Anliegen des Anwenders), werden die vielen Einzelroutinen von zentraler Stelle aus über variable Vektoren angesprungen. Zwar unterstützt Basic mit den Befehlen ON GOTO und ON GOSUB Mehrfachverzweigungen, diese sind aber bei der hier in Frage kommenden sehr großen Zahl von Sprungzielen schlecht einzusetzen. Die beiden Maschinenprogramme GS und GT (siehe auch Tabelle 2) erlauben dagegen (Unterprogramm-) Sprünge zu Zeilen, deren Nummer ein beliebiger arithmetischer Ausdruck oder eine Variable, der Sprungvektor, sein kann.

Die Tastenkommandos werden nun sehr schnell und auf einfache Weise interpretiert. Der ASCII-Code A der zuletzt gedrückten Taste und die Nummer M des gerade aktiven Menüs (0 bis 7) werden als Indizes in eine Sprungvektor-Tabelle V% verwendet. Der gefundene Eintrag ist die Zeilennummer der Routine, die den Tastendruck bearbeitet. Diese Routine wird als Unterprogramm aufgerufen. Für diesen Vorgang benötigt man nur einen einzigen Befehl: SYSGS,V%(M,A) (Zeile 1550).

Allen nicht definierten Tasten wird über die Tabelle der Sprungvektor 0 zugeordnet. Er führt in das leere Unterprogramm ganz am Anfang des Listings. Die Sprung-Tabelle wird über die Inhalte der DATA-Zeilen ab 9500 im Programm-Setup besetzt. Vierstellige DATAs stehen für Zeilennummern, dreistellige für ASCII-Codes, zweistellige wählen ein Menü (dabei bedeutet zum Beispiel M1 »Menü Nr. 1« und MA »alle Menüs«) und einstellige sind ASCII-Zeichen. »ENDE« schließt die Tabellen-DATAs ab. Der Programmteil ab Zeile 8110 interpretiert diese DATAs und besetzt die Tabelle.

Eine Ergänzung des Programms um weitere Funktionen gestattet sich mit diesem Konzept sehr einfach. Man schreibt ein Unterprogramm und trägt die Zeilennummer des Unterprogramm-Starts zusammen mit der Taste, durch die das Unterprogramm aufgerufen werden soll, in die Tabelle ein. Die Hardcopys entstanden auf diese Weise: Ab Zeile 7900 stand eine Hardcopy-Routine (im Listing hier nicht enthalten). Durch:
9760 DATA MA,7900,Z
wurde bewirkt, daß in allen Menüs durch den Tastendruck Z ein Hardcopy-Ausdruck ausgelöst werden konnte.

Im Sound-Editor werden noch weitere Sprungvektoren angewendet: Eine f-Taste führt grundsätzlich in die Dummy-Routine (leeres Unterprogramm) oder in eine der vier Zeilen 2550, 2560, 2570, 2580. Dort wird ein aktueller numerischer Parameterwert erhöht oder erniedrigt. Anschließend wird in eine der 21 Routinen ab Zeile 2600 weiterverzweigt. Die zugehörige Vektorvariable ist PV. Die 21 Einzelroutinen bringen den geänderten Parameter an seinen Platz in ein SID- oder ein Modulator-Register und geben ihn zudem an der richtigen Stelle am Bildschirm aus.

Der Parameter-Vektor PV wird von Parameter-Anwahl-Routinen gesetzt, von denen es in jedem Menü für jede Parameterart eine gibt, zum Beispiel »Filterfrequenz« wählen, Zeile 5200. Der Aufruf einer Parameter-Anwahl-Routine wird durch die Cursor-Tasten ausgelöst und geschieht durch einen weiteren Vektor AV. Beispiel: »Parameterwahl durch Cursor«, Zeile 5130. Von diesen Cursor-Routinen gibt es fünf Ausführungen, für jedes Menü, in denen die Cursor-Tasten wirksam sind, eine.

Einige Maschinenprogramme machen den Sound-Editor schneller und übersichtlicher. Erwähnt seien dabei Unterprogramme zum Zugriff auf Doppelbytes von Basic aus (»DEEK« und »DOKE«). Das lästige Berechnen von Low- und High-Byte entfällt so. Der »PRINT AT«-Befehl gestattet es, den Cursor an jede Stelle des Bildschirms zu positionieren, um dort etwas auszugeben. Die Tabellen 1 bis 4 sollen eine kleine Dokumentation des Sound-Editors darstellen. Der Leser, der das Programm selbst modifizieren oder erweitern will (zum Beispiel andere oder mehr Farben), findet dort die gesuchte Information.

Das Tabellen- und Vektor-Konzept wird einen weiteren Ausbau des Programms sehr erleichtern. Das schon einmal angekündigte Sequenzer-Programm wird bis zum Ende dieses Kurses den Sound-Editor zu einem kompletten Synthesizer-Programm erweitern, mit dem man dann auch virtuose Melodien mit Modulator-Sounds spielen lassen kann.

(Thomas Krätzig/ev)

Real

A	allgemeine Verwendung, ASCII-Codes
AD	allgemeine Adresse
AN	Auswahl-Nummer bei EG (0=A 1=D 2=S 3=R 4=EGA)
AV	Auswahl-Vektor bei Parameterwahl durch Cursor
B	allgemeine Verwendung
CL	Startadresse CLEAR (Zeile 0-14)
DO	Startadresse DOKE
EG	Status des Soft-EG (0 = abgekoppelt, —1 = angekoppelt)
F	allgemeiner Farbcode
F1	Code für Normalfarbe
F2	Code für Hervorhebungsfarbe
FA	Basisadresse Farb-RAM
GS	Startadresse berechnetes GOSUB
GT	Startadresse berechnetes GOTO
I,J,K	allgemeine Laufvariablen
L	Stringlänge
LN	Nummer des aktuellen LFO
M	aktuelles Untermenü (0 bis 7)
MO	Basisadresse Modulator
NS	Sound-Nummer (0 bis 23)
O	Oktave (temporär) 0 bis 5
OF	Tonhöhen-Offset (temporär) -49 bis +50
PA	Parameter-Zieladresse
PB	Farb-RAM-Adresse des aktuellen Parameterfeldes
PM	Maximum+1 des aktuellen (numerischen) Parameters
PR	Startadresse PRINT AT
PV	Sprungvektor zur Weiterverarbeitung des aktuellen Parameters
PW	Wert des aktuellen Parameters
P1	Schrittweite langsam
P2	Schrittweite schnell
RF	RES/FILT Inhalt von SID $17
S	Spalte allgemein
SA	Sound-Adresse (Anfang eines Parameterblocks)
SI	Basisadresse SID
SN	Nummer der aktuellen Stimme
SO	Basisadresse Sounds
T	Tonnummer (temporär) 0 — 11
X,Y	allgemeine Verwendung
Z	Zeile allgemein
ZN	Zeilennummer allgemein

Strings

A$	Eingabezeichen, allgemeine Verwendung
F1$	Steuerzeichen Normalfarbe
F2$	Steuerzeichen Hervorhebungsfarbe
DN$	Dateiname

Zahlen-Arrays

AD(2)	Attack/Decay für 3 Stimmen
BF(2)	Basisfrequenz für 3 Stimmen
C(2)	Controlbyte (GATE aus) für 3 Stimmen
C1(2)	Controlbyte (GATE an) für 3 Stimmen
O(2)	Oktaven für 3 Stimmen 0 bis 3
OF(2)	Tonhöhen-Offset für 3 Stimmen -49 bis +50
S(2)	Status der Stimmen (0 = inaktiv, —1 = aktiv)
SR(2)	Sustain/Release für 3 Stimmen
T(2)	Tonhöhen-Nummer für 3 Stimmen 0 bis 11
TH(24)	relative Tonhöhen
TN%(255)	Tonnummern (0 bis 24) TN%(A) = Nummer des Tones bei Tastendruck CHR$(A)
V%(7,255)	Sprungvektoren

String-Arrays

BL$(6)	0 bis 6 Blanks
KF$(3)	LFO-Kurvenformen grafisch
KV$(3)	Namen der LFO-Kurvenformen
MD$(3)	Namen der Modulationsquellen-Modi: RESET, HOLD, RUN
TN$(11)	Namen der Töne: C, C#, D, D#, E, F, F#, G, G#, A, A#, H

Tabelle 1. Variablen des Sound-Editors

Name	Startadresse	Aufruf	Funktion
GS	828	SYSGS, Zeilennummer	Berechnetes GOSUB
GT	857	SYSGT, Zeilennummer	Berechnetes GOTO
PR	869	SYSPR, Zeile, Spalte, Printliste	PRINT AT: Positioniert den Cursor an der Position Zeile (0-24), Spalte (0-39) und geht über zum Basic-PRINT
DO	897	SYSDO, Adresse, Wert	DOKE: Doppelbyte-POKE Schreibt einen 16-Bit-Wert (0-65536) ab Adresse in den Speicher, niederwertiges Byte zuerst
USR	937	USR (Adresse)	DEEK: Doppelbyte-PEEK Liefert 16-Bit-Wert der Speicherstellen-Adresse und Adresse +1
CL	959	SYSCL	CLEAR: Löscht Bildschirm Zeilen 0-14
NA	50229	SYSNA, Adresse	NAME: Ausgabe von 10 Zeichen, deren ASCII-Codes ab Adresse im Speicher stehen. Dient zur Ausgabe von Soundnamen
GE	50245	SYSGE, Adresse	GET: Belegt SID und MODULATOR mit Parametersatz, der ab Adresse im Speicher steht (siehe Tabelle 4)
PU	50276	SYSPU, Adresse	PUT: Kopiert Parameter von MODULATOR (aber keine SID-Parameter!) in Parameterblock ab Adresse (siehe Tabelle 4)

Alle hier aufgeführten Unterprogramme ermitteln ihre Parameter mit Hilfe des Basic-Interpreters. Es sind daher beliebige arithmetische Ausdrücke (beziehungsweise String-Ausdrücke bei PR) möglich. Es wird ein »Illegal Quantity Error« ausgelöst, wenn die Parameter ihren zugelassenen Bereich (0-255 beziehungsweise 0-65535) überschreiten.

Tabelle 2. Unterprogramme zum Sound-Editor in Maschinensprache

Zero-Page	$02	
	$9B	von MODULATOR benutzt
	$FB-$FE	
	$9E-$9F	von NA, GE und PU benutzt
USR-Vektor	$311-$312	belegt mit Startadresse von DEEK ($03A9)
Kassetten-Puffer	$33C-$3D4	Maschinenprogramme GS, GT, PR, DO, USR (DEEK), CL
Basic	$0800-$08FF	(obere Speichergrenze heruntergesetzt)
Sounds	$9000-$9A07	24 Parameter-Blöcke à 107 Byte
Freier RAM-Bereich	$C000-$C070	MODULATOR-Parameter
	$C071-$C074	MODULATOR-interne Variablen
	$C075-$C422	MODULATOR-Code
	$C423-$C473	Maschinenprogramme NA, GE, PU

Tabelle 3. Speicherbelegung durch den Sound-Editor

Es werden 24 Parameter-Blöcke der Länge 107 Byte im Bereich $9000 bis $9A08 verwaltet.

Struktur eines Parameter-Blocks

Relativadresse dezimal	Relativadresse hex	Parameter	Variable in Basic (falls vorhanden)
0 — 72	$00 — $48	Modulator-Parameter	keine
73 — 74	$49 — $4A	Basisfrequenz Stimme 1	BF(2)
75	$4B	Status Stimme 1 (0 = inaktiv, 1 = aktiv)	S(2)
76	$4C	unbenutzt	
77	$4D	Control Register Stimme 1	C(2)
78	$4E	Attach/Decay Stimme 1	AD(2)
79	$4F	Sustain/Release Stimme 1	SR(2)
80 — 86	$50 — $56	analog Stimme 2 (7 Byte)	
87 — 93	$57 — $5D	analog Stimme 3 (7 Byte)	
94	$5E	Resonanz/Filter	RF
95	$5F	Status des Soft-EG (0 = abgekoppelt, 1 = angekoppelt)	EG
96	$60	Status Sustain (0 = aus, 1 = an)	SU
97 — 106	$61 — $6A	Name des Sounds (10 Byte)	keine

Tabelle 4. Speicherung der Sound-Parameter

```
0 IF A>1 THEN RETURN :REM DUMMY-ROUTINE      <139>
1000 REM *****************************        <102>
1005 REM ***                        ***       <123>
1010 REM ***    SOUND - EDITOR      ***       <253>
1015 REM ***                        ***       <133>
1020 REM ***  FUER DIE PARAMETER    ***       <047>
1025 REM *** VON SID UND MODULATOR  ***       <001>
1030 REM ***                        ***       <149>
1035 REM *** T. KRAETZIG      1985  ***       <084>
1040 REM ***                        ***       <159>
1045 REM *****************************        <148>
1050 REM                                      <173>
1055 REM FALLS VON MENUE 7 GEKOMMEN,          <071>
1060 REM DORTHIN ZURUECK                      <248>
1065 IF M=7 THEN 7665                         <109>
1070 REM                                      <193>
1120 REM MODULATOR LADEN                      <014>
1130 IF A=0 THEN A=1:LOAD"MODULATOR",8,1      <112>
1140 REM                                      <007>
1150 REM DIMENSIONIERUNGEN                    <020>
1155 POKE 56,144:CLR:REM B.-GRENZE $9000      <066>
1160 DIM V%(7,255),TN%(255),TH(24)            <060>
1165 DIM BF(2),T(2),O(2),OF(2),S(2)           <105>
1170 DIM C1(2),C(2),AD(2),SR(2)               <141>
1175 DIM BL$(6),KV$(3),KF$(6),MD$(3)          <156>
1180 DIM TN$(11)                              <159>
1185 REM                                      <052>
1190 GOSUB 8000  :REM INITIALISIERUNGEN       <203>
1200 GOSUB 4000  :REM HAUPTMENUE              <202>
1210 REM                                      <077>
1500 REM==============================        <211>
1510 REM HAUPTSCHLEIFE, SPRUNGVERTEILER       <252>
1520 REM==============================        <231>
1530 GET A$:IF A$="" THEN 1530                <149>
1540 A=ASC(A$)                                <121>
1550 SYS GS,V%(M,A) :REM BERECHN. GOSUB       <108>
1560 GOTO 1530                                <112>
2000 REM==============================        <201>
2010 REM NOTEN SPIELEN                        <004>
2020 REM==============================        <221>
2025 POKE 650,0                               <126>
2030 F=TH(TN%(A)):AD=MO+72:X=PEEK(AD)         <047>
2035 IF S(0) THEN SYS DO,MO   ,F*BF(0):POK
     E SI+4 ,C1(0)                            <140>
2040 IF S(1) THEN SYS DO,MO+7 ,F*BF(1):POK
     E SI+11,C1(1)                            <163>
2050 IF S(2) THEN SYS DO,MO+14,F*BF(2):POK
     E SI+18,C1(2)                            <230>
2090 IF EG THEN POKE AD,X OR 1                <113>
2110 IF SU THEN POKE 650,128:RETURN           <225>
2120 IF PEEK(203)<>64 AND PEEK(198)=0 THEN
      2120                                    <003>
2130 POKE AD,X AND 254                        <068>
2140 POKE SI+4 ,C(0)                          <101>
2150 POKE SI+11,C(1):POKE SI+18,C(2)          <016>
2160 POKE 650,128:RETURN                      <057>
2500 REM==============================        <191>
2510 REM NUMERISCHE PARAMETER AENDERN,        <035>
2520 REM VERTEILER EINZELROUTINEN             <093>
2530 REM==============================        <221>
2550 PW=PW+P2:IF PW>=PM THEN PW=PW-PM         <134>
2555 SYS GT,PV                                <015>
2560 PW=PW+P1:IF PW>=PM THEN PW=PW-PM         <144>
2565 SYS GT,PV                                <026>
2570 PW=PW-P1:IF PW< 0  THEN PW=PW+PM         <125>
2575 SYS GT,PV                                <036>
2580 PW=PW-P2:IF PW< 0  THEN PW=PW+PM         <136>
2585 SYS GT,PV                                <046>
2600 REM------------------------------        <052>
2605 REM LFOF                                 <237>
2610 SYS DO,PA,PW:L=LEN(STR$(PW))             <005>
2620 SYS PR,3+LN,4,BL$(6-L);F2$;PW            <158>
2625 RETURN                                   <217>
2630 REM------------------------------        <082>
2635 REM LFOP                                 <021>
2640 POKE PA,PW:L=LEN(STR$(PW))               <093>
2650 SYS PR,3+LN,11,BL$(4-L);F2$;PW           <232>
2655 RETURN                                   <247>
2660 REM------------------------------        <112>
2665 REM LFOA                                 <036>
2670 POKE PA,PW:L=LEN(STR$(PW))               <123>
2680 SYS PR,3+LN,16,BL$(4-L);F2$;PW           <011>
2685 RETURN                                   <021>
2690 REM------------------------------        <142>
2695 REM LFO KURVE                            <142>
2700 IF A>136 OR A<133 THEN RETURN            <029>
2705 B=A-133  :REM B=0,1,2,3                  <164>
2710 POKE PA,(PEEK(PA)AND 231)OR(B*8)         <085>
2715 SYS PR,3+LN,21,F2$;KV$(B)                <114>
2717 SYS PR,6   ,34,F2$;KF$(B):RETURN         <239>
2720 REM------------------------------        <172>
2725 REM LFO MODUS                            <167>
2730 IF A>135 OR A<133 THEN RETURN            <058>
2735 B=A-132  :REM B=1,2,3                    <101>
2740 POKE PA,(PEEK(PA)AND 249)OR(B*2)         <118>
2745 SYS PR,3+LN,28,F2$;MD$(B):RETURN         <079>
2750 REM------------------------------        <202>
2755 REM EG A D S R EGA (AN=0-4)              <141>
2760 POKE PA,PW:L=LEN(STR$(PW))               <213>
2765 SYS PR,7,S,BL$(4-L);F2$;PW:RETURN        <212>
2770 REM------------------------------        <222>
2775 REM EG FORM                              <048>
2780 B=-(A>134):REM F1,F3 B=0 F5,F7 B=1       <200>
2785 POKE PA,(PEEK(PA)AND 247)OR(B*8)         <167>
2790 SYS PR,7,29,F2$;CHR$(43+2*B)             <118>
2795 RETURN                                   <131>
2810 REM------------------------------        <006>
2815 REM EG MODUS                             <172>
2820 IF A>135 OR A<133 THEN RETURN            <149>
2825 B=A-132  :REM B=1,2,3                    <192>
2830 POKE PA,(PEEK(PA)AND 249)OR(B*2)         <209>
2835 SYS PR,7,33,F2$;MD$(B):RETURN            <102>
2840 REM------------------------------        <037>
2845 REM FILTER FREQUENZ                      <237>
2850 POKE PA,PW AND 7:POKE PA+1,INT(PW/8)     <009>
2855 L=LEN(STR$(PW))                          <000>
2860 SYS PR,5,2,BL$(5-L);F2$;PW:RETURN        <018>
2865 REM------------------------------        <062>
2870 REM FILTER RESONANZ                      <006>
2875 RF=(RF AND 15)+16*PW:POKE PA,RF          <024>
2880 L=LEN(STR$(PW))                          <025>
2885 SYS PR,5,9,BL$(3-L);F2$;PW:RETURN        <048>
2890 REM------------------------------        <087>
2895 REM FILTER SCHALTER                      <005>
2900 IF A>136 OR A<133 THEN RETURN            <230>
2905 B=A-133  :REM B=0,1,2,3                  <109>
2910 X=2↑B:X1=255-X:REM MASKEN                <049>
2915 Y=RF AND X                               <024>
2920 RF=(RF AND X1)OR(X-Y):POKE PA,RF         <097>
2925 Y=-128*(Y=0):AD=1024+(5+B)*40+15         <117>
2930 FOR I=AD TO AD+5                         <134>
2935 : POKE I,(PEEK(I)AND 127)OR Y            <199>
2940 NEXT:RETURN                              <209>
2945 REM------------------------------        <142>
2950 REM FILTER MODUS                         <110>
2955 IF A>136 OR A<133 THEN RETURN            <029>
2960 B=A-129  :REM B=4,5,6,7                  <185>
2965 X=2↑B:X1=255-X:REM MASKEN                <104>
2970 PW=PEEK(PA):Y=PW AND X                   <149>
2975 POKE PA,(PW AND X1)OR(X-Y)               <139>
2980 Y=-128*(Y=0):AD=1024+(1+B)*40+25         <169>
2985 FOR I=AD TO AD+3                         <187>
2990 : POKE I,(PEEK(I)AND 127)OR Y            <254>
2995 NEXT:RETURN                              <008>
3000 REM------------------------------        <197>
3005 REM LAUTSTAERKE                          <156>
3010 POKE PA,(PEEK(PA) AND 240) OR PW         <178>
3015 L=LEN(STR$(PW))                          <160>
3020 SYS PR,5,31,BL$(3-L);F2$;PW:RETURN       <226>
3050 REM------------------------------        <247>
3052 REM BASISFREQUENZ                        <104>
3054 IF A<>134 THEN 3064                      <053>
3056 REM F3 OFFSET ERHOEHEN                   <024>
3058 OF=OF+1:IF OF<=50 THEN 3110              <068>
3062 OF=-49:GOTO 3076                         <243>
3064 IF A<>135 THEN 3074                      <065>
3066 REM F5 OFFSET ERNIEDRIGEN                <002>
3068 OF=OF-1:IF OF>=-49 THEN 3110             <000>
3072 OF=50:GOTO 3084                          <074>
3074 IF A<>133 THEN 3082                      <073>
3076 REM TONNUMMER ERHOEHEN                   <178>
3078 T=T+1:IF T<=11 THEN 3110                 <147>
3080 T=0:GOTO 3090                            <217>
3082 IF A<>136 THEN RETURN                    <021>
3084 REM TONNUMMER ERNIEDRIGEN                <152>
3086 T=T-1:IF T>=0 THEN 3110                  <104>
3088 T=11:GOTO 3096                           <025>
3090 REM OKTAVE ERHOEHEN                      <197>
3092 O=O+1:IF O<=5 THEN 3110                  <101>
3094 O=0:GOTO 3110                            <219>
3096 REM OKTAVE ERNIEDRIGEN                   <169>
3098 O=O-1:IF O>=0 THEN 3110                  <101>
3100 O=5:T=10                                 <255>
3110 REM BASISFREQUENZ BERECHNEN              <045>
3112 IF T=11 AND O=5 THEN T=0:O=0             <084>
3114 BF=440*2↑(O-4+(T-9)/12+OF/1200)          <005>
3116 O(SN)=O:T(SN)=T:OF(SN)=OF                <008>
3118 BF(SN)=BF*17.0327                        <252>
3120 SYS DO,MO+7*SN,F*BF(SN)                  <251>
3122 REM ANZEIGEN                             <030>
3124 SYS PR,3+SN,3,F2$;O                      <219>
3126 SYS PR,3+SN,2,TN$(T)                     <033>
3128 SYS PR,3+SN,5,RIGHT$(" "+STR$(OF),3)     <204>
3130 RETURN                                   <212>
3150 REM------------------------------        <092>
3155 REM PULSWEITE STIMME                     <127>
3160 SYS DO,PA,PW                             <037>
3165 SYS PR,3+SN,9,F2$;RIGHT$("{2SPACE}"+S
     TR$(PW),4)                               <223>
3170 RETURN                                   <252>
3175 REM------------------------------        <117>
3180 REM A                                    <072>
3185 AD(SN)=(AD(SN)AND 15) OR PW*16           <237>
3190 POKE PA,AD(SN)                           <077>
3195 SYS PR,3+SN,14,F2$;RIGHT$(STR$(PW),2)    <057>
3200 RETURN                                   <026>
3205 REM------------------------------        <147>
3210 REM D                                    <105>
3215 AD(SN)=(AD(SN)AND 240) OR PW             <040>
3220 POKE PA,AD(SN)                           <107>
3225 SYS PR,3+SN,17,F2$;RIGHT$(STR$(PW),2)    <090>
3230 RETURN                                   <056>
3235 REM------------------------------        <177>
3240 REM S                                    <150>
3245 SR(SN)=(SR(SN)AND 15) OR PW*16           <105>
3250 POKE PA,SR(SN)                           <169>
3255 SYS PR,3+SN,20,F2$;RIGHT$(STR$(PW),2)    <114>
3260 RETURN                                   <086>
3265 REM------------------------------        <207>
3270 REM R                                    <179>
3275 SR(SN)=(SR(SN)AND 240) OR PW             <164>
3280 POKE PA,SR(SN)                           <199>
3285 SYS PR,3+SN,23,F2$;RIGHT$(STR$(PW),2)    <147>
3290 RETURN                                   <116>
3300 REM------------------------------        <242>
3305 REM CONTROL (BIT 7-S)                    <140>
3310 IF A<>133 AND A<>134 THEN 3335           <034>
3315 REM BIT SETZEN                           <070>
3320 C(SN)=C(SN) OR 2↑(7-S)                   <054>
3325 C1(SN)=C(SN) OR 1                        <055>
3330 SYS PR,3+SN,26+S,F2$;"1":GOTO 3355       <149>
3335 REM BIT RUECKSETZEN                      <213>
3340 C(SN)=C(SN) AND 255-2↑(7-S)              <145>
3345 C1(SN)=C(SN) OR 1                        <076>
3350 SYS PR,3+SN,26+S,F2$;"0"                 <021>
3355 REM CONTROL-BYTE ZUM SID                 <021>
3360 X=C(SN):IF SU THEN X=C1(SN)              <208>
3365 POKE SI+7*SN+4,X:RETURN                  <189>
3370 REM------------------------------        <057>
3375 REM PORTA                                <081>
3380 POKE PA,PW:L=LEN(STR$(PW))               <068>
3385 SYS PR,3+SN,34,BL$(4-L);F2$;PW           <214>
3390 RETURN                                   <217>
3500 REM==============================        <171>
3510 REM STIMME (DES-)AKTIVIEREN              <109>
3520 REM==============================        <191>
3525 SYS PR,15,10,F2$;"STIMME 1,2,3"          <053>
3530 GET A$:IF A$=""THEN 3530                 <111>
3540 IF A$="1"OR A$="2"OR A$="3"THEN 3560     <232>
3550 GOTO 3590                                <069>
3560 I=VAL(A$)-1 :S(I)=NOT S(I)               <181>
3570 IF S(I) THEN PRINT CHR$(18);             <115>
3580 SYS PR,21,31+2*I,F1$;A$;CHR$(146);       <201>
3590 SYS PR,15,10,F1$;"STIMME 1,2,3"          <118>
3595 RETURN                                   <167>
3600 REM==============================        <016>
3610 REM SOFTWARE-EG AUSLOESEN                <138>
3620 REM==============================        <036>
3630 POKE 650,64                              <004>
3640 AD=MO+72:X=PEEK(AD):POKE AD,X OR 1       <195>
3650 IF PEEK(203)<>64 AND PEEK(198)=0 THEN
      3650                                    <012>
3660 POKE AD,X AND 254:POKE 650,128:RETURN    <063>
3700 REM==============================        <116>
3710 REM SOFTWARE-EG AN-/ABKOPPELN            <198>
3720 REM==============================        <136>
3730 EG=NOT EG                                <018>
3740 IF EG THEN PRINT CHR$(18);               <189>
3750 SYS PR,23,31,F1$;"SOFT-EG{RVOFF}":RET
     URN                                      <170>
3800 REM==============================        <216>
3810 REM SUSTAIN-MODUS EIN/AUS                <079>
3820 REM==============================        <236>
3830 SU=NOT SU                                <174>
3840 IF SU THEN PRINT CHR$(18);               <062>
3850 SYS PR,22,31,F1$;"SUSTAIN{RVOFF}"        <120>
3860 IF NOT SU THEN 2140                      <108>
3870 RETURN                                   <187>
4000 REM==============================        <161>
4010 REM HAUPTMENUE                           <068>
4020 REM==============================        <181>
4030 M=0:PV=0:SYS CL:PRINT "{HOME}";F1$       <037>
4040 PRINT" ACCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCS"                                 <126>
4045 PRINT" B{RVSON,9SPACE}M O D U L A T O
      R{10SPACE,RVOFF}B"                      <152>
4050 PRINT" B{RVSON,5SPACE}THOMAS KRAETZIG
     {7SPACE}1985{5SPACE,RVOFF}B"             <234>
4055 PRINT" ZCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCX"                                 <153>
4100 SYS PR,6,1,F2$;"SH V";F1$;" STIMMEN"     <189>
4105 SYS PR,7,1,F2$;"{3SPACE}F";F1$;" FILT
     ER/LAUTSTAERKE"                          <053>
4110 SYS PR,8,1,F2$;"{3SPACE}L";F1$;" LFO"    <232>
4115 SYS PR,9,1,F2$;"{3SPACE}H";F1$;" HUEL
     LKURVE (SOFT-EG)"                        <086>
4120 SYS PR,10,1,F2$;"{3SPACE}K";F1$;" KSV
     "                                        <045>
4125 SYS PR,11,1,F2$;"SH S";F1$;" SOUNDS"     <190>
4130 SYS PR,12,1,F2$;"{3SPACE}D";F1$;" DIS
     K"                                       <105>
4135 RETURN                                   <197>
4200 REM------------------------------        <122>
4210 REM BILD ZEILEN 15-23                    <235>
4220 REM------------------------------        <142>
4240 SYS PR,15,1, F2$;"V{2SPACE}1,2,3"        <090>
4245 SYS PR,15,10,F1$;"STIMME 1,2,3           <229>
4250 SYS PR,15,26,F2$;"S"                     <170>
4255 SYS PR,15,32,F1$;"SUSTAIN"               <127>
4260 SYS PR,16,1, F2$;"SH SPACE"              <050>
4270 SYS PR,16,10,F1$;"EG (AB-)KOPPELN"       <010>
4280 SYS PR,16,26,F2$;"SPACE"                 <226>
4290 SYS PR,16,32,F1$;"EG GATE"               <041>
4292 SYS PR,17,1, F2$;"X"                     <164>
4294 SYS PR,17,10,F1$;"HAUPTMENUE"            <121>
4300 SYS PR,19,0,F2$;                         <017>
4310 FOR I=1 TO 4                             <107>
4320 : PRINT"{SPACE,RVSON,SPACE}B B B B B
     B B B B B B B B{SPACE,RVOFF}":NEXT       <003>
4330 PRINT"{SPACE,RVSON}QBWBEBRBTBYBUBIBOB
     PB@B*B↑B;B="                             <238>
4340 SYS PR,19,0,;                            <157>
4350 FOR I=1 TO 2                             <145>
4360 : PRINT"{SPACE,RIGHT,SPACE,RIGHT,SPAC
     E,3RIGHT,SPACE,RIGHT,SPACE,RIGHT,SPAC
     E,3RIGHT,SPACE,RIGHT,SPACE,3RIGHT,SPA
     CE,RIGHT,SPACE,RIGHT,SPACE}":NEXT        <188>
4370 PRINT"{SPACE,RIGHT}2{RIGHT}3{3RIGHT}5
     {RIGHT}6{RIGHT}7{3RIGHT}9{RIGHT}0{3RI
     GHT}-{RIGHT}£{RIGHT}C"                   <012>
4380 SYS PR,19,31,F1$;"AKTIV:"                <146>
4390 FOR I=0 TO 2                             <185>
4400 : A$=RIGHT$(STR$(I+1),1)                 <066>
4410 : GOSUB 3570:NEXT I                      <230>
4420 GOSUB 3840:GOSUB 3740                    <070>
4430 RETURN                                   <237>
4500 REM==============================        <151>
4505 REM UNTERMENUE STIMME                    <016>
4510 REM==============================        <161>
4515 M=1:PV=3050:SYS CL:PRINT"{HOME,RVOFF}
     ";F1$                                    <055>
4520 PRINT"{5SPACE}F{4SPACE}PW{3SPACE}A{2S
     PACE}D{2SPACE}S{2SPACE}R CONTROL PORT
     A"                                       <084>
4525 PRINT" ACCCCCCRCCCCRCCRCCRCCRCCRCCCCC
     CCRCCCCS"                                <238>
4530 FOR I=1 TO 3                             <071>
4535 PRINT" B{6SPACE}B{4SPACE}B{2SPACE}B{2
     SPACE}B{2SPACE}B{2SPACE}B{7SPACE}B{4S
     PACE}B"                                  <119>
4540 NEXT I                                   <152>
4545 PRINT" ZCCCCCCECCCCECCECCECCECCERRRRR
     RRECCCCX"                                <144>
4550 SYS PR,7,26,"BBBBBB7CCSYNC"              <122>
4555 SYS PR,8,26,"BBBBB7CCRING"               <177>
4560 SYS PR,9,1,F2$;"F1{2SPACE}++";F1$        <205>
4565 SYS PR,9,26,"BBBB7CCTEST"                <010>
4570 SYS PR,10,1,F2$;"F3{2SPACE}+";F1$        <214>
4575 SYS PR,10,26,"BBB7CCTRIAN"               <184>
4580 SYS PR,11,1,F2$;"F5{2SPACE}-";F1$        <229>
4585 SYS PR,11,26,"BB7CCSAWTOOTH"             <252>
4590 SYS PR,12,1,F2$;"F7{2SPACE}--";F1$       <031>
4595 SYS PR,12,26,"B7CCSQUARE"                <157>
4600 SYS PR,13,26,"7CCNOISE"                  <142>
4602 REM PARAMETER ANZEIGEN                   <140>
4604 A$=F2$:F2$=F1$                           <227>
4606 FOR SN=0 TO 2                            <233>
4608 : T=T(SN):O=O(SN):OF=OF(SN)              <028>
4610 : GOSUB 3122:REM F                       <178>
4612 : PW=(USR(MO+7*SN+2)):GOSUB 3165         <062>
4614 : PW=INT(AD(SN)/16)  :GOSUB 3195         <213>
4616 : PW=AD(SN)AND 15     :GOSUB 3225        <204>
4618 : PW=INT(SR(SN)/16)  :GOSUB 3255         <246>
4620 : PW=SR(SN)AND 15     :GOSUB 3285        <246>
4622 : REM CONTROL-BYTE (7 BIT)               <210>
4624 : B=C(SN):A=128                          <077>
4626 : SYS PR,3+SN,26,F1$;"0000000"           <114>
4628 : FOR S=7 TO 1 STEP -1                   <119>
4630 : IF(B AND A)THEN SYS PR,3+SN,33-S,"1
     "                                        <050>
4632 : A=A/2:NEXT S                           <134>
4634 : PA=MO+7*SN+4:PW=PEEK(PA)               <136>
4636 : GOSUB 3370:REM PORTA                   <017>
4638 NEXT SN                                  <083>
4640 F2$=A$                                   <197>
4642 SN=0:AV=4700:SYS GS,AV:RETURN            <002>
4650 REM------------------------------        <062>
4652 REM PARAMETERWAHL DURCH CURSOR           <238>
4654 REM HOCH                                 <241>
4656 SN=SN-1:IF SN<0 THEN SN=2                <135>
4658 GOSUB 5815:SYS GT,AV                     <218>
4660 REM RUNTER                               <181>
4662 SN=SN+1:IF SN>2 THEN SN=0                <138>
4664 GOSUB 5815:SYS GT,AV                     <224>
4666 REM RECHTS                               <164>
4668 IF AV<>4880 THEN 4674                    <036>
4670 S=S+1:IF S>6 THEN AV=4910                <064>
4672 GOTO 4678                                <180>
4674 AV=AV+30:IF AV>4910 THEN AV=4700         <215>
4676 IF AV=4880 THEN S=0                      <218>
4678 GOSUB 5815:SYS GT,AV                     <238>
4680 REM LINKS                                <106>
4682 IF AV<>4880 THEN 4688                    <055>
4684 S=S-1:IF S<0 THEN AV=4850                <078>
4686 GOTO 4692                                <190>
4688 AV=AV-30:IF AV<4700 THEN AV=4910         <232>
4690 IF AV=4880 THEN S=6                      <238>
4692 GOSUB 5815:SYS GT,AV                     <252>
4700 REM------------------------------        <112>
4702 REM BASISFREQUENZ WAEHLEN                <229>
4704 PB=FA+40*(SN+3)+2                        <162>
4706 FOR I=PB TO PB+5:POKE I,F2:NEXT          <017>
4708 T=T(SN):O=O(SN):OF=OF(SN)                <070>
4710 PA=MO+7*SN:PV=3050:RETURN                <167>
4730 REM------------------------------        <142>
4732 REM PULSWEITE WAEHLEN                    <227>
4734 PB=FA+40*(SN+3)+9                        <199>
4736 FOR I=PB TO PB+3:POKE I,F2:NEXT          <045>
4738 PA=MO+7*SN+2:PW=USR(PA):PM=4096          <005>
4740 P1=1:P2=50:PV=3150:RETURN                <240>
4760 REM------------------------------        <172>
4762 REM ATTACK WAEHLEN                       <247>
4764 PB=FA+40*(SN+3)+14                       <017>
```

Listing »Sound-Editor«
(Fortsetzung auf Seite 139)
Beachten Sie bitte Seite 77.







```
4766 POKE PB,F2:POKE PB+1,F2                   <095>
4768 PA=SI+7*SN+5:PW=INT(AD(SN)/16)            <194>
4770 PM=16:P1=1:P2=1:PV=3175:RETURN            <209>
4790 REM------------------------------         <202>
4792 REM DECAY WAEHLEN                         <195>
4794 PB=FA+40*(SN+3)+17                        <050>
4796 POKE PB,F2:POKE PB+1,F2                   <125>
4798 PA=SI+7*SN+5:PW=AD(SN)AND 15              <219>
4800 PM=16:P1=1:P2=1:PV=3205:RETURN            <233>
4820 REM------------------------------         <232>
4822 REM SUSTAIN WAEHLEN                       <162>
4824 PB=FA+40*(SN+3)+20                        <074>
4826 POKE PB,F2:POKE PB+1,F2                   <155>
4828 PA=SI+7*SN+6:PW=INT(SR(SN)/16)            <031>
4830 PM=16:P1=1:P2=1:PV=3235:RETURN            <010>
4850 REM------------------------------         <006>
4852 REM RELEASE WAEHLEN                       <154>
4854 PB=FA+40*(SN+3)+23                        <107>
4856 POKE PB,F2:POKE PB+1,F2                   <185>
4858 PA=SI+7*SN+6:PW=SR(SN)AND 15              <056>
4860 PM=16:P1=1:P2=1:PV=3265:RETURN            <043>
4880 REM------------------------------         <037>
4882 REM CONTROL-BYTE WAEHLEN                  <058>
4884 PB=FA+40*(SN+3)+26+S                      <138>
4886 POKE PB,F2                                <246>
4888 PA=SI+7*SN+4:PV=3300:RETURN               <054>
4910 REM------------------------------         <067>
4912 REM PORTA WAEHLEN                         <092>
4914 PB=FA+40*(SN+3)+35                        <171>
4916 FOR I=0 TO 2:POKE PB+I,F2:NEXT            <127>
4918 PA=MO+7*SN+4:PW=PEEK(PA):PM=256           <145>
4920 P1=1:P2=10:PV=3370:RETURN                 <165>
5000 REM==============================         <141>
5005 REM UNTERMENUE FILTER/LAUT                <098>
5010 REM==============================         <151>
5015 M=2:PV=0:SYS CL:PRINT"{HOME,RVOFF}";F
     1$                                        <150>
5020 PRINT" FILTER/LAUTSTAERKE"                <198>
5025 PRINT" ACCCCCRCCCCCRCCCCCCCCRCCCCCCCR
     CCCCS"                                    <206>
5030 PRINT" BFILTFB RES BSCHALTERB MODUS B
     LAUTB"                                    <149>
5035 PRINT" QCCCCC+CCCCC+CCCCCCCC+CCCCCCC+
     CCCCW"                                    <124>
5040 PRINT" B{5SPACE}B{5SPACE}B FILT1{2SPA
     CE}B{3SPACE}LP{2SPACE}B{4SPACE}B"         <040>
5045 PRINT" QCCCCCFCCCCCW FILT2{2SPACE}B{3
     SPACE}BP{2SPACE}QCCCCX"                   <123>
5050 PRINT" B{11SPACE}B FILT3{2SPACE}B{3SP
     ACE}HP{2SPACE}B"                          <172>
5055 PRINT" B{11SPACE}B FILTEX B{2SPACE}3O
     FF B"                                     <145>
5060 PRINT" ZCCCCCCCCCCCFCCCCCCCCFCCCCCCCX
     "                                         <078>
5065 PRINT F2$;" F1";F1$;"{4SPACE}++{6SPAC
     E}FILT1{6SPACE}LP{4SPACE}++"              <004>
5070 PRINT F2$;" F3";F1$;"{4SPACE}+{7SPACE
     }FILT2{6SPACE}BP{4SPACE}+"                <172>
5075 PRINT F2$;" F5";F1$;"{4SPACE}-{7SPACE
     }FILT3{6SPACE}HP{4SPACE}-"                <190>
5080 PRINT F2$;" F7";F1$;"{4SPACE}-{7SPACE
     }FILTEX{4SPACE}3OFF{3SPACE}--"            <210>
5085 PW=(PEEK(MO+21)AND 7)+8*PEEK(MO+22)       <135>
5090 SYS PR,5, 7-LEN(STR$(PW)),PW              <071>
5095 PW=INT(RF/16)                             <005>
5100 SYS PR,5,12-LEN(STR$(PW)),PW              <125>
5102 IF RF AND 1 THEN SYS PR,5,15,"{RVSON}
     FILT1{SPACE,RVOFF}"                       <082>
5104 IF RF AND 2 THEN SYS PR,6,15,"{RVSON}
     FILT2{SPACE,RVOFF}"                       <087>
5106 IF RF AND 4 THEN SYS PR,7,15,"{RVSON}
     FILT3{SPACE,RVOFF}"                       <093>
5108 IF RF AND 8 THEN SYS PR,8,15,"{RVSON}
     TILTEX{RVOFF}"                            <220>
5110 PW=PEEK(MO+23)                            <032>
5112 IF PW AND 16 THEN SYS PR,5,25,"{RVSON
     ,SPACE}LP{SPACE,RVOFF}"                   <222>
5114 IF PW AND 32 THEN SYS PR,6,25,"{RVSON
     ,SPACE}BP{SPACE,RVOFF}"                   <213>
5116 IF PW AND 64 THEN SYS PR,7,25,"{RVSON
     ,SPACE}HP{SPACE,RVOFF}"                   <227>
5118 IF PW AND 128 THEN SYS PR,8,25,"{RVSO
     N}3OFF{RVOFF}"                            <141>
5120 PW=PEEK(MO+23)AND 15                      <064>
5122 SYS PR,5,34-LEN(STR$(PW)),PW              <152>
5125 AV=5200:SYS GS,AV:RETURN                  <038>
5130 REM------------------------------         <032>
5135 REM PARAMETERWAHL DURCH CURSOR            <211>
5140 REM RECHTS                                <128>
5145 AV=AV+50:IF AV>5400 THEN AV=5200          <169>
5150 GOTO 5165                                 <140>
5155 REM LINKS                                 <071>
5160 AV=AV-50:IF AV<5200 THEN AV=5400          <187>
5165 GOSUB 5815:A=PB-FA-200                    <207>
5170 IF A<>15 AND A<>25 THEN SYS GT,AV         <058>
5175 FOR J=1 TO 3:PB=PB+40:GOSUB 5815:NEXT     <068>
5180 PB=PB-120:SYS GT,AV                       <154>
5200 REM------------------------------         <102>
5205 REM FILTERFREQUENZ ANWAEHLEN              <193>
5210 PB=FA+203                                 <120>
5215 FOR I=PB TO PB+3:POKE I,F2:NEXT           <014>
5220 PA=MO+21:PW=PEEK(PA)+8*PEEK(PA+1)         <168>
5225 PM=2048:P1=1:P2=50:PV=2850:RETURN         <052>
5250 REM------------------------------         <152>
5255 REM RESONANZ ANWAEHLEN                    <045>
5260 PB=FA+210                                 <168>
5265 POKE PB,F2:POKE PB+1,F2                   <084>
5270 PA=SI+23:PW=INT(RF/16)                    <221>
5275 PM=16:P1=1:P2=1:PV=2875:RETURN            <210>
5300 REM------------------------------         <202>
5305 REM SCHALTER ANWAEHLEN                    <069>
5310 PB=FA+215                                 <223>
5315 FOR I=0 TO 120 STEP 40:FOR J=0 TO 5       <135>
5320 : POKE PB+I+J,F2:NEXT:NEXT                <066>
5325 PA=SI+23:PW=RF AND 15                     <015>
5330 PV=2900:RETURN                            <209>
5350 REM------------------------------         <252>
5355 REM MODUS ANWAEHLEN                       <169>
5360 PB=FA+225                                 <018>
5365 FOR I=0 TO 120 STEP 40:FOR J=0 TO 3       <183>
5370 : POKE PB+I+J,F2:NEXT:NEXT                <116>
5375 PA=MO+23                                  <001>
5380 PV=2955:RETURN                            <014>
5400 REM------------------------------         <047>
5405 REM LAUTSTAERKE ANWAEHLEN                 <153>
5410 PB=FA+232                                 <067>
5415 POKE PB,F2:POKE PB+1,F2                   <235>
5420 PA=MO+23:PW=PEEK(PA) AND 15               <123>
5425 PM=16:P1=1:P2=1:PV=3010:RETURN            <087>
5500 REM==============================         <131>
5505 REM UNTERMENUE LFO                        <014>
5510 REM==============================         <141>
5515 M=3:SYS CL:PRINT"{HOME,RVOFF}";F1$;       <006>
5520 PRINT"{3SPACE}ACCCCCCRCCCCRCCCCRCCCCC
     CRCCCCCS"                                 <179>
5525 PRINT"{3SPACE}B LFOF BLFOPBLFOABKURVE
     BMODUSB"                                  <162>
5530 PRINT" AC+CCCCCC+CCCC+CCCC+CCCCCC+CCC
     CC+CCCCS"                                 <230>
5535 FOR Z=3 TO 9                              <081>
5540 : SYS PR,Z,1,"B B{6SPACE}B{4SPACE}B{4
     SPACE}B{6SPACE}B{5SPACE}B{4SPACE}B"       <150>
5545 NEXT Z                                    <154>
5550 SYS PR,10,1,"ZCFCCCCCCFCCCCFCCCCFCCCC
     CCFCCCCCFCCCCX                            <165>
5555 PRINT F2$;" F1";F1$;"{2SPACE}++{5SPAC
     E}++{3SPACE}++{2SPACE}TRIAN{2SPACE}RE
     SET"                                      <075>
5560 PRINT F2$;" F3";F1$;"{2SPACE}+{6SPACE
     }+{4SPACE}+{3SPACE}SAWUP{2SPACE}HOLD"     <135>
5565 PRINT F2$;" F5";F1$;"{2SPACE}-{6SPACE
     }-{4SPACE}-{3SPACE}SAWDWN RUN"            <166>
5570 PRINT F2$;" F7";F1$;"{2SPACE}--{5SPAC
     E}--{3SPACE}--{2SPACE}SQUARE"             <060>
5575 FOR LN=0 TO 6                             <179>
5580 : Z=3+LN                                  <158>
5585 : SYS PR,Z,2,CHR$(48+LN)                  <056>
5590 : PW=USR(MO+32+5*LN)      :REM LFOF       <070>
5595 : SYS PR,Z,10-LEN(STR$(PW)),PW            <203>
5600 : PW=PEEK(MO+34+5*LN)     :REM LFOP       <103>
5605 : SYS PR,Z,15-LEN(STR$(PW)),PW            <218>
5610 : PW=PEEK(MO+35+5*LN)     :REM LFOA       <099>
5615 : SYS PR,Z,20-LEN(STR$(PW)),PW            <224>
5620 : PW=PEEK(MO+36+5*LN)     :REM LFOC       <112>
5625 : SYS PR,Z,21,KV$((PW AND 24)/8)          <209>
5630 : SYS PR,Z,28,MD$((PW AND  6)/2)          <151>
5640 NEXT LN                                   <058>
5645 LN=0:AV=5830:SYS GS,AV:RETURN             <239>
5650 REM------------------------------         <042>
5655 REM PARAMETERWAHL DURCH CURSOR            <221>
5670 REM HOCH                                  <237>
5675 LN=LN-1:IF LN<0 THEN LN=6                 <110>
5680 GOSUB 5815:SYS GT,AV                      <220>
5685 REM RUNTER                                <186>
5690 LN=LN+1:IF LN>6 THEN LN=0                 <122>
5695 GOSUB 5815:SYS GT,AV                      <235>
5700 REM RECHTS                                <178>
5705 AV=AV+30:IF AV>5950 THEN AV=5830          <236>
5710 GOSUB 5810:SYS GT,AV                      <245>
5715 REM LINKS                                 <121>
5720 AV=AV-30:IF AV<5830 THEN AV=5950          <254>
5725 GOSUB 5810:SYS GT,AV                      <004>
5800 REM------------------------------         <192>
5805 REM AKTUELES PARAMETERFELD FARBE 1        <253>
5810 SYS PR,6,34,"{4SPACE}"                    <109>
5815 FOR I=PB TO PB+6:POKE I,F1:NEXT:RETUR
     N                                         <050>
5820 REM------------------------------         <212>
5825 REM LFOF ANWAEHLEN                        <032>
5830 PB=FA+40*(LN+3)+4                         <007>
5835 FOR I=PB TO PB+5:POKE I,F2:NEXT           <126>
5840 PA=MO+32+5*LN:PW=USR(PA):PM=65536         <183>
5845 P1=1:P2=200:PV=2610:RETURN                <114>
5850 REM------------------------------         <242>
5855 REM LFOP ANWAEHLEN                        <072>
5860 PB=FA+40*(LN+3)+11                        <083>
5865 FOR I=PB TO PB+3:POKE I,F2:NEXT           <154>
5870 PA=MO+34+5*LN:PW=PEEK(PA):PM=256          <118>
5875 P1=1:P2=10 :PV=2640:RETURN                <098>
5880 REM------------------------------         <016>
5885 REM LFOA WAEHLEN                          <200>
5890 PB=FA+40*(LN+3)+16                        <119>
5895 FOR I=PB TO PB+3:POKE I,F2:NEXT           <185>
5900 PA=MO+35+5*LN:PW=PEEK(PA):PM=256          <150>
5905 P1=1:P2=10 :PV=2670:RETURN                <132>
5910 REM------------------------------         <047>
5915 REM LFO KURVE WAEHLEN                     <051>
5920 PB=FA+40*(LN+3)+21                        <113>
5925 FOR I=PB TO PB+5:POKE I,F2:NEXT           <217>
5930 PA=MO+36+5*LN:PV=2700                     <179>
5935 B=(PEEK(PA) AND 24)/8:GOTO 2717           <189>
5940 REM------------------------------         <077>
5945 REM LFO MODUS WAEHLEN                     <076>
5950 PB=FA+40*(LN+3)+28                        <182>
5955 FOR J=PB TO PB+4:POKE J,F2:NEXT           <248>
5960 PA=MO+36+5*LN                             <118>
5965 PV=2730:RETURN                            <080>
6000 REM==============================         <121>
6010 REM UNTERMENUE EG                         <180>
6020 REM==============================         <141>
6030 M=4:SYS CL:PRINT"{HOME,RVOFF}";F1$        <209>
6040 PRINT" SOFTWARE-EG":PRINT:PRINT           <086>
6050 PRINT" ACCCCRCCCCRCCCCRCCCCRCCCCRCCCC
     CRCCCCCS"                                 <250>
6060 PRINT" B{2SPACE}A B{2SPACE}D B{2SPACE
     }S B{2SPACE}R B EGAB FORMBMODUSB"         <099>
6070 PRINT" QCCCC+CCCC+CCCC+CCCC+CCCC+CCCC
     C+CCCCCW"                                 <004>
6080 PRINT" B{4SPACE}B{4SPACE}B{4SPACE}B{4
     SPACE}B{4SPACE}B{5SPACE}B{5SPACE}B"       <196>
6090 PRINT" ZCCCCFCCCCFCCCCFCCCCFCCCCFCCCC
     CFCCCCCX"                                 <040>
6100 SYS PR,10,1,F2$;"F1":PRINT" F3"           <099>
6110 PRINT" F5":PRINT" F7";F1$                 <183>
6120 SYS PR,10,13,"++{18SPACE}RESET"           <165>
6130 SYS PR,11,13,"+{15SPACE}+{3SPACE}HOLD
     "                                         <084>
6140 SYS PR,12,13,"-{15SPACE}-{3SPACE}RUN"     <049>
6150 SYS PR,13,13,"--"                         <072>
6160 FOR I=0 TO 4                              <172>
6170 : PW=PEEK(MO+67+I)                        <126>
6180 : SYS PR,7,6+5*I-LEN(STR$(PW)),PW         <157>
6190 NEXT I                                    <017>
6210 PW=PEEK(MO+72)                            <117>
6220 SYS PR,7,29,CHR$(43+(PW AND 8)/4)         <179>
6230 SYS PR,7,33,MD$((PW AND 6)/2)             <145>
6240 AV=6310:AN=0:SYS GT,AV                    <108>
6300 REM------------------------------         <182>
6305 REM A D S R EGA (AN=0-4) WAEHLEN          <241>
6310 PB=FA+282+5*AN:S=2+5*AN                   <116>
6315 FOR I=PB TO PB+3:POKE I,F2:NEXT           <094>
6320 PA=MO+67+AN  :PW=PEEK(PA)                 <046>
6325 PM=256:IF AN=0 THEN PM=129                <129>
6330 P1=1:P2=10:PV=2760:RETURN                 <046>
6335 REM------------------------------         <217>
6340 REM EG FORM WAEHLEN                       <047>
6345 PB=FA+280+27:POKE PB+2,F2                 <230>
6350 PA=MO+72:PV=2780:RETURN                   <003>
6370 REM------------------------------         <252>
6375 REM EG MODUS WAEHLEN                      <166>
6380 PB=FA+280+33                              <035>
6385 FOR I=PB TO PB+4:POKE I,F2:NEXT           <165>
6390 PA=MO+72:PV=2820:RETURN                   <038>
6395 REM------------------------------         <021>
6400 REM PARAMETERWAHL DURCH CURSOR            <201>
6405 REM LINKS                                 <046>
6410 IF AV<>6310 THEN 6420                     <230>
6415 AN=AN-1:IF AN>=0 THEN 6430                <047>
6420 AV=AV-35:AN=4                             <207>
6425 IF AV<6310 THEN AV=6380                   <146>
6430 GOSUB 5815:SYS GT,AV                      <205>
6435 REM RECHTS                                <148>
6440 IF AV<>6310 THEN 6450                     <007>
6445 AN=AN+1:IF AN<=4 THEN 6460                <085>
6450 AV=AV+35                                  <061>
6455 IF AV>6380 THEN AV=6310:AN=0              <089>
6460 GOSUB 5815:SYS GT,AV                      <235>
6500 REM==============================         <111>
6510 REM UNTERMENUE KSV                        <018>
6520 REM==============================         <131>
6530 M=5:PV=0:PRINT"{RVOFF}";:SYS CL           <220>
6540 SYS PR,1,1,F1$;"KREUZSCHIENEN-VERTEIL
     ER"                                       <162>
6550 SYS PR,3,1,"E{4SPACE}LFO"                 <065>
6560 SYS PR,4,1,"G6543210"                     <210>
6570 SYS PR,5,1,F2$;"@@@@@@@@"                 <040>
6580 A$=F2$+"H"+F1$+"{8SPACE}"+F2$+"I "+F1
     $                                         <149>
6590 PRINT A$;"F1";F2$;"{5SPACE}CURSORTAST
     EN"                                       <206>
6600 PRINT A$;"F2"                             <214>
6610 PRINT A$;"F3"                             <225>
6620 PRINT A$;"P1";F2$;"{5SPACE}F1 ";F1$;"
     VERBINDUNG"                               <009>
6630 PRINT A$;"P2"                             <254>
6640 PRINT A$;"P3";F2$;"{5SPACE}F7 ";F1$;"
     KEINE VERBINDUNG"                         <145>
6650 PRINT A$;"FILT"                           <191>
6660 PRINT A$;"LAUT"                           <209>
6670 PRINT F2$;" TTTTTTTT"                     <244>
6680 FOR Z=0 TO 7:GOSUB 6730:NEXT              <029>
6685 Z=0:S=7:POKE 55296+40*(6+Z)+8-S,F2        <142>
6690 RETURN                                    <202>
6700 REM------------------------------         <072>
6710 REM KSV-BYTE Z ANZEIGEN                   <223>
6720 REM------------------------------         <092>
6730 B=PEEK(MO+24+Z):A=128                     <227>
6740 SYS PR,6+Z,1,F1$;"{RVSON}00000000"        <073>
6750 IF B=0 THEN RETURN                        <092>
6760 FOR S=7 TO 0 STEP -1                      <152>
6770 IF(B AND A)THEN SYS PR,6+Z,8-S,"{RVSO
     N}1";                                     <055>
6780 A=A/2:NEXT S                              <184>
6790 PRINT F1$;:RETURN                         <215>
6800 REM------------------------------         <172>
6810 REM CURSORBEWEGUNGEN                      <008>
6820 REM------------------------------         <192>
6830 POKE 55296+40*(6+Z)+8-S,F1:REM AUS        <228>
6840 Z=(Z-1) AND 7:GOTO 6910:REM HOCH          <201>
6850 POKE 55296+40*(6+Z)+8-S,F1                <070>
6860 Z=(Z+1) AND 7:GOTO 6910:REM RUNTER        <154>
6870 POKE 55296+40*(6+Z)+8-S,F1                <090>
6880 S=(S+1) AND 7:GOTO 6910:REM LINKS         <065>
6890 POKE 55296+40*(6+Z)+8-S,F1                <110>
6900 S=(S-1) AND 7:GOTO 6910:REM RECHTS        <158>
6910 POKE 55296+40*(6+Z)+8-S,F2:REM AN         <219>
6915 RETURN                                    <172>
6920 REM------------------------------         <037>
6925 REM KSV BIT Z;S SETZEN                    <060>
6930 REM------------------------------         <047>
6935 AD=MO+24+Z                                <025>
6940 POKE AD,PEEK(AD) OR 2↑S                   <250>
6950 POKE 1024+40*(6+Z)+8-S,177:RETURN         <087>
6960 REM------------------------------         <077>
6965 REM KSV BIT Z,S SETZEN                    <100>
6970 REM------------------------------         <087>
6975 AD=MO+24+Z                                <065>
6980 POKE AD,PEEK(AD) AND (255-2↑S)            <185>
6990 POKE 1024+40*(6+Z)+8-S,176:RETURN         <126>
7000 REM==============================         <101>
7005 REM UNTERMENUE SOUNDS                     <235>
7010 REM==============================         <111>
7015 M=6:PV=0:SYS CL:PRINT"{HOME,RVOFF}";F
     2$                                        <115>
7020 SYS PR,1,8,"A{11SPACE}B{11SPACE}C"        <190>
7025 PRINT F1$;"{3SPACE}ACCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCS"                       <131>
7030 FOR I=1 TO 8                              <026>
7035 SYS PR,2+I,1,F2$;CHR$(48+I);F1$" B"       <218>
7040 SYS PR,2+I,38,"B":NEXT                    <177>
7045 SYS PR,11,3,"ZCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCX"                             <141>
7050 SYS PR,12,4,F2$;"A-C 1-8"                 <098>
7055 SYS PR,12,13,F1$;"SOUND WAEHLEN"          <220>
7060 SYS PR,13,1,F2$;"←{2SPACE}A-C 1-8"        <201>
7065 SYS PR,13,13,F1$"SOUND ABSPEICHERN"       <204>
7070 SYS PR,14,1,F2$;"N"                       <124>
7075 SYS PR,14,13,F1$;"NAMEN EINGEBEN (RET
     URN)"                                     <066>
7080 FOR S=0 TO 2:FOR Z=0 TO 7                 <033>
7085 : SYS PR,3+Z,4+S*12,;                     <120>
7090 : SYS NA,SO+107*(8*S+Z)+97                <150>
7095 NEXT:NEXT:IF NS<0 THEN Z=0:S=0:RETURN     <115>
7100 Z=NS AND 7:S=INT(NS/8):GOTO 7180          <200>
7110 REM------------------------------         <227>
7115 REM SOUND WAEHLEN (A,B,C)                 <113>
7120 X=A-65 :REM SPALTE 0,1,2                  <201>
7122 SYS PR,12,13,F2$;"{RVOFF}SOUND WAEHLE
     N"                                        <178>
7125 GET A$:IF A$="" THEN 7125                 <140>
7130 Y=ASC(A$)-49:REM ZEILE                    <214>
7135 IF Y<0 OR Y>7 THEN 7142                   <039>
7140 GOSUB 7150:GOSUB 7400                     <235>
7142 SYS PR,12,13,F1$;"{RVOFF}SOUND WAEHLE
     N"                                        <197>
7144 RETURN                                    <145>
7146 REM------------------------------         <007>
7150 REM ALTES NAMENFELD NORMAL                <100>
7155 NS=8*S+Z                                  <156>
7160 : SYS PR,3+Z,4+S*12,F1$;"{RVOFF}";        <111>
7165 : SYS NA,SO+107*NS+97                     <246>
7170 REM NEUES NAMENFELD HERVORHEBEN           <239>
7175 S=X:Z=Y:NS=8*S+Z                          <231>
7180 : SYS PR,3+Z,4+S*12,F2$;"{RVOFF}";        <133>
7185 : SYS NA,SO+107*NS+97                     <011>
7190 RETURN                                    <192>
7200 REM------------------------------         <030>
7205 REM SOUND ABSPEICHERN                     <125>
7210 SYS PR,13,13,F2$;"{RVOFF}SOUND ABSPEI
     CHERN"                                    <044>
7215 GET A$:IF A$="" THEN 7215                 <231>
7220 X=ASC(A$)-65:REM SPALTE                   <126>
7225 IF X<0 OR X>2 THEN 7250                   <123>
7230 GET A$:IF A$="" THEN 7230                 <243>
7235 Y=ASC(A$)-49:REM ZEILE                    <064>
7240 IF Y<0 OR Y>7 THEN 7250                   <145>
7245 GOSUB 7150:GOSUB 7450:GOTO 7260           <236>
7250 SYS PR,13,13,F1$;"{RVOFF}SOUND ABSPEI
     CHERN"                                    <083>
7255 RETURN                                    <001>
7260 SYS PR,13,13,F1$;"{RVOFF}SOUND ABSPEI
     CHERN"                                    <093>
7300 REM------------------------------         <162>
7305 REM NAMEN EINGEBEN (Z,S,NS)               <215>
7310 SYS PR,14,13,F2$;"NAMEN EINGEBEN (RET
     URN)"                                     <047>
7315 X=12*S+4:Y=Z+3:AD=1024+40*Y+X             <194>
7320 SA=SO+107*NS+97:I=0                       <170>
7325 POKE AD+I,PEEK(AD+I) OR 128               <202>
7330 GET A$:IF A$="" THEN 7330                 <088>
7335 A=ASC(A$):IF A<32 OR A>95 THEN 7370       <216>
7340 REM ZEICHEN AUSGEBEN                      <167>
7345 SYS PR,Y,X+I,F2$;A$;:POKE SA+I,A          <113>
7350 I=I+1                                     <241>
```

Listing »Sound-Editor«
(Fortsetzung von Seite 136)

```
7355 IF I>9 THEN I=9:GOTO 7325                <004>
7360 IF I<0 THEN I=0                          <101>
7365 GOTO 7325                                <059>
7370 IF A=29 THEN POKE AD+I,PEEK(AD+I) AND
      127:GOTO 7350                           <023>
7375 IF A=157 THEN POKE AD+I,PEEK(AD+I) AN
     D 127:I=I-1:GOTO 7355                    <173>
7380 IF A<>13 THEN 7330                       <248>
7385 POKE AD+I,PEEK(AD+I) AND 127             <004>
7390 SYS PR,14,13,F1$;"NAMEN EINGEBEN (RET
     URN)"                                    <126>
7395 RETURN                                   <141>
7400 REM--------------------------------      <006>
7402 REM PARAMETERSATZ NS HOLEN               <143>
7404 SA=SO+NS*107:SYS GE,SA                   <163>
7406 FOR I=0 TO 2                             <140>
7408 : AD=SA+7*I                              <231>
7410 : BF(I)=USR(AD+73)                       <189>
7412 : X=BF(I)/17.0327/440                    <202>
7414 : Y=LOG(X)/LOG(2) :REM X=2↑Y             <239>
7416 : Y=Y+4+9/12+49/1200                     <010>
7418 : O(I)=INT(Y):Y=(Y-O(I))*12              <220>
7420 : T(I)=INT(Y)                            <081>
7422 : OF(I)=INT((Y-T(I))*100-48.5)           <053>
7424 : S(I)=(PEEK(AD+75)=1)                   <216>
7426 : A$=CHR$(49+I):GOSUB 3570               <126>
7428 : C(I)=PEEK(AD+77) AND 254               <228>
7430 : C1(I)=C(I) OR 1                        <219>
7432 : AD(I)=PEEK(AD+78)                      <225>
7434 : SR(I)=PEEK(AD+79)                      <004>
7436 NEXT I                                   <244>
7438 RF=PEEK(SA+94)                           <051>
7440 EG=(PEEK(SA+95)=1):GOSUB 3740            <243>
7442 SU=(PEEK(SA+96)=1):GOSUB 3840            <019>
7444 RETURN                                   <191>
7450 REM--------------------------------      <057>
7452 REM PARAMETERSATZ NS ABSPEICHERN         <112>
7454 SA=SO+NS*107:SYS PU,SA                   <239>
7456 FOR SN=0 TO 2                            <247>
7458 : AD=SA+7*SN                             <114>
7460 : SYS DO,AD+73,BF(SN)                    <023>
7462 : POKE AD+75,-S(SN)                      <203>
7464 : POKE AD+77,C(SN)                       <020>
7466 : POKE AD+78,AD(SN)                      <089>
7468 : POKE AD+79,SR(SN)                      <124>
7470 NEXT SN                                  <110>
7472 POKE SA+94,RF                            <083>
7474 POKE SA+95,-EG                           <245>
7476 POKE SA+96,-SU                           <020>
7478 RETURN                                   <225>
7500 REM================================      <091>
7505 REM UNTERMENUE DISKETTE                  <098>
7510 REM================================      <101>
7515 M=7:PV=0:SYS CL:PRINT"{HOME}";F1$        <215>
7520 PRINT" DISK"                             <133>
7525 SYS PR,5,4,F2$;"F1"                      <065>
7530 SYS PR,5,7,F1$;"SOUNDS LADEN"            <017>
7535 SYS PR,7,4,F2$;"F3"                      <079>
7540 SYS PR,7,7,F1$;"SOUNDS ABSPEICHERN"      <221>
7545 RETURN                                   <036>
7600 REM--------------------------------      <207>
7605 REM SOUNDS LADEN                         <161>
7610 SYS PR,5,7,F2$;"SOUNDS LADEN"            <098>
7615 SYS PR,10,4,F1$;"DATEINAME ";            <018>
7620 DN$="SOUNDS{14SPACE}"                    <105>
7625 SYS PR,10,16,DN$                         <040>
7630 SYS PR,10,14,;:INPUT DN$                 <037>
7635 OPEN 8,8,8,DN$+",P,R":CLOSE 8            <107>
7640 OPEN 1,8,15:INPUT#1,A,A$,X,Y:CLOSE 1     <188>
7645 SYS PR,12,4,"{29SPACE}"                  <153>
7650 IF A=0 THEN 7660                         <039>
7655 SYS PR,12,3,A;A$;X;Y:GOTO 7625           <253>
7660 A=0:LOAD DN$,8,1                         <112>
7665 SYS PR,5,7,F1$;"SOUNDS LADEN"            <152>
7670 NS=-1:A=211:GOTO 1550                    <145>
7700 REM--------------------------------      <052>
7705 REM SOUNDS ABSPEICHERN                   <198>
7710 SYS PR,7,7,F2$;"SOUNDS ABSPEICHERN"      <137>
7715 SYS PR,10,4,F1$;"DATEINAME ";            <119>
7720 DN$="SOUNDS{14SPACE}"                    <206>
7725 SYS PR,10,16,DN$                         <141>
7730 SYS PR,10,14,;:INPUT DN$                 <138>
7735 SYS 50198:REM MODULATOR AUS              <099>
7740 REM BEREICH $8000-$8A08 AUF DISK         <000>
7745 OPEN 1,8,1,"@:"+DN$                      <058>
7750 POKE 252,0:POKE 253,144:REM $9000        <214>
7755 POKE 780,252:REM AKKU                    <089>
7760 POKE 781,8  :REM X-REG = $08             <254>
7765 POKE 782,154:REM Y-REG = $9A             <121>
7770 SYS 65496   :REM KERNAL SAVE             <217>
7775 CLOSE 1                                  <078>
7780 OPEN 1,8,15:INPUT#1,A,A$,X,Y:CLOSE 1     <073>
7785 IF A=0 THEN 7795                         <184>
7790 SYS PR,12,4,A;A$;X;Y:GOTO 7725           <135>
7795 SYS PR,12,4,"{30SPACE}"                  <048>
7800 SYS PR,7,7,F1$;"SOUNDS ABSPEICHERN"      <226>
7805 SYS 50185:RETURN:REM MODULATOR AN        <019>
8000 REM================================      <081>
8010 REM VORBESETZUNG, INITIALISIERUNG        <066>
8020 REM================================      <101>
8030 PRINT"{CLR,12DOWN,7SPACE}";              <244>
8035 PRINT"BITTE 30 SEKUNDEN WARTEN"          <104>
8040 REM--------------------------------      <137>
8045 REM MS-PROGRAMME IN KASSETTENPUF.        <244>
8050 FOR I=0 TO 22   :REM KERNAL-GOSUB        <120>
8055 POKE 828+I,PEEK(43139+I):NEXT            <180>
8060 FOR I=0 TO 129:READ X:POKE 851+I,X:NE
     XT                                       <066>
8065 POKE 785,169:POKE 786,3:REM USR          <063>
8070 REM--------------------------------      <167>
8072 REM MS-PROGRAMME AB $C423                <048>
8074 FOR I=50211 TO 50291                     <195>
8076 READ X:POKE I,X:NEXT                     <228>
8078 REM--------------------------------      <175>
8080 REM NAMEN DER TOENE                      <003>
8082 FOR I=0 TO 11:READ TN$(I):NEXT           <064>
8100 REM--------------------------------      <197>
8110 REM SPRUNGVEKTOREN                       <169>
8120 READ A$:L=LEN(A$)                        <116>
8130 ON L GOTO 8150,8220,8280,8310            <003>
8150 REM L=1  ASCII-ZEICHEN                   <218>
8160 A=ASC(A$)                                <110>
8170 IF M>=0 THEN V%(M,A)=ZN:GOTO 8120        <137>
8180 FOR I=0 TO 7:V%(I,A)=ZN:NEXT             <108>
8190 GOTO 8120                                <113>
8200 IF LEFT$(A$,1)<>"M" THEN 8120            <213>
8220 REM L=2  MENUE-INDIKATOR                 <178>
8230 IF LEFT$(A$,1)<>"M" THEN 8120            <243>
8240 A$=RIGHT$(A$,1)                          <035>
8250 IF A$="A" THEN M=-1 :GOTO 8120           <113>
8260 M=VAL(A$)              :GOTO 8120        <076>
8280 REM L=3  ASCII-WERT                      <123>
8290 A=VAL(A$) :GOTO 8170                     <099>
8310 REM L=4  ZEILENNUMMER ODER ENDE          <163>
8320 IF A$="ENDE" THEN 8340                   <248>
8330 ZN=VAL(A$) :GOTO 8120                    <237>
8340 REM--------------------------------      <150>
8350 REM TONNUMMERN                           <064>
8360 A$="Q2W3ERST6Y7UI9O0P@-*£↑{HOME};="      <120>
8370 FOR I=0 TO 24                            <104>
8380 : A=ASC(MID$(A$,I+1,1))                  <045>
8390 : TN%(A)=I                               <084>
8400 : FOR J=0 TO 7:V%(J,A)=2000:NEXT         <127>
8410 NEXT I                                   <165>
8450 REM--------------------------------      <037>
8455 REM FREQUENZEN DER TONLEITER             <075>
8460 FOR I=0 TO 24:TH(I)=2↑(I/12):NEXT        <075>
8465 REM--------------------------------      <052>
8470 REM KONSTANTEN                           <203>
8472 GS=828:GT=857:PR=869:DO=897:CL=959       <208>
8474 NA=50229:GE=50245:PU=50276               <139>
8480 SI=54272:MO=49152:FA=55296               <126>
8482 SO=36864                                 <162>
8484 F1$=CHR$(154):F1=14                      <012>
8490 F2$=CHR$(5)  :F2=1                       <123>
8495 KV$(0)="TRIAN ":KF$(0)="[illegible]"     <166>
8500 KV$(1)="SAWUP ":KF$(1)="[illegible]"     <189>
8505 KV$(2)="SAWDWN":KF$(2)="@M@M"            <226>
8510 KV$(3)="SQUARE":KF$(3)="OLOL"            <014>
8515 MD$(0)="RESET" :MD$(1)="RESET"           <253>
8520 MD$(2)="HOLD " :MD$(3)="RUN{2SPACE}"     <028>
8525 BL$(0)="":BL$(1)=" ":BL$(2)="{2SPACE}
     "                                        <096>
8530 BL$(3)="{3SPACE}":BL$(4)="{4SPACE}"      <006>
8535 BL$(5)="{5SPACE}":BL$(6)="{6SPACE}"      <015>
8540 REM--------------------------------      <127>
8545 REM SOUNDPARAMETER VORBESETZUNGEN        <124>
8550 FOR SN=0 TO 2                            <097>
8555 : POKE SI+7*SN+5,10 :AD(SN)=10           <087>
8560 : POKE SI+7*SN+6,138:SR(SN)=138          <243>
8565 : O(SN)=3+SN:OF(SN)=0:T(SN)=0            <070>
8570 : BF=440*2↑(O(SN)-4+(T(SN)-9)/12+OF(S
     N)/1200)                                 <122>
8572 : BF(SN)=BF*17.0327                      <153>
8575 :  C(SN)=64            :C1(SN)=65        <232>
8580 :  S(SN)=(SN=0)                          <170>
8585 : SYS DO,MO+7*SN+2,2048 :REM PW          <081>
8590 : POKE  MO+7*SN+4,0    :REM PORTA        <211>
8595 NEXT SN                                  <215>
8600 POKE  SI+23,0:RF=0  :REM RES/FILT        <028>
8605 SYS DO,MO+21,0       :REM FILTERFR.      <117>
8610 POKE  MO+23,15       :REM MOD/LAUT       <165>
8615 FOR I=24 TO 72 :REM KSV,LFO,EG AUS       <034>
8620 : POKE MO+I,0:NEXT                       <069>
8625 EG=0            :REM EG ABKOPPELN        <049>
8630 SU=0            :REM SUSTAIN AUS         <058>
8635 REM--------------------------------      <222>
8640 REM SOUNDS VORBESETZEN                   <163>
8645 FOR NS=0 TO 23                           <243>
8650 : GOSUB 7450                             <130>
8655 : AD=SO+107*NS+97                        <172>
8660 : FOR I=AD TO AD+9:POKE I,46:NEXT        <168>
8665 NEXT NS:NS=0                             <218>
8700 REM--------------------------------      <031>
8870 REM FARBEN, BILD, MODULATOR              <047>
8880 POKE 53280,14:POKE 53281,6               <050>
8890 PRINT F1$;"{RVOFF}";:POKE 650,128        <143>
8900 GOSUB 4200 :REM BILD ZEILEN 15-23        <220>
8910 SYS MO+1033:REM MODULATOR AKTIV.         <200>
8920 A=2:RETURN                               <231>
9000 REM================================      <061>
9010 REM DATAS                                <081>
9020 REM================================      <081>
9030 REM MASCHINENPROGRAMME                   <056>
9060 DATA 032,089,003,076,174,167,032         <071>
9070 DATA 253,174,032,138,173,032,247         <080>
9080 DATA 183,076,163,168,032,253,174         <099>
9090 DATA 032,158,183,138,072,032,253         <099>
9100 DATA 174,032,158,183,138,168,104         <117>
9110 DATA 170,024,032,240,255,032,253         <104>
9120 DATA 174,076,160,170,032,253,174         <129>
9130 DATA 032,138,173,032,247,183,165         <142>
9140 DATA 020,133,158,165,021,133,159         <142>
9150 DATA 032,253,174,032,138,173,032         <152>
9160 DATA 247,183,160,000,120,165,020         <151>
9170 DATA 145,158,200,165,021,145,158         <177>
9180 DATA 088,096,032,247,183,160,000         <190>
9190 DATA 120,177,020,133,099,200,177         <194>
9200 DATA 020,088,133,098,162,144,056         <213>
9210 DATA 076,073,188,169,032,162,000         <222>
9220 DATA 157,000,004,157,000,005,202         <202>
9230 DATA 208,247,162,087,157,000,006         <239>
9240 DATA 202,016,250,096                     <164>
9250 REM--------------------------------      <072>
9260 REM MASCHINENPROGRAMME AB $C423          <162>
9270 DATA 032,253,174,032,138,173,032         <017>
9280 DATA 247,183,165,020,133,158,165         <039>
9290 DATA 021,133,159,096,032,035,196         <046>
9300 DATA 160,000,177,158,032,210,255         <043>
9310 DATA 200,192,010,208,246,096,032         <054>
9320 DATA 035,196,160,000,177,158,153         <076>
9330 DATA 000,192,200,192,073,208,246         <075>
9340 DATA 177,158,153,183,211,200,192         <095>
9350 DATA 094,208,246,177,158,141,023         <112>
9360 DATA 212,096,032,035,196,160,000         <103>
9370 DATA 185,000,192,145,158,200,192         <121>
9380 DATA 073,208,246,096                     <062>
9400 REM--------------------------------      <222>
9410 REM  NAMEN DER TOENE                     <058>
9420 DATA C,C#,D,D#,E,F,F#,G,G#,A,A#,H        <061>
9500 REM--------------------------------      <067>
9510 REM DATAS FUER SPRUNGVERTEILER           <010>
9520 DATA MA,4000,X,3500,V,3600,032           <006>
9530 DATA MA,3700,160,3800,S                  <159>
9540 DATA MA,2550,133,2560,134                <242>
9550 DATA MA,2570,135,2580,136,4500,214       <188>
9555 DATA MA,5500,L,6000,H,6500,K             <203>
9560 DATA MA,5000,F,7000,211,7500,D           <012>
9565 DATA M1,4654,145,4660,017                <008>
9570 DATA M1,4666,029,4680,157                <024>
9580 DATA M2,5160,157,5145,029                <022>
9600 DATA M3,5675,145,5690,017                <053>
9610 DATA M3,5720,157,5705,029                <057>
9620 DATA M4,6410,157,6440,029                <062>
9630 DATA M5,6830,145,6850,017                <078>
9640 DATA M5,6870,157,6890,029                <102>
9650 DATA M5,6935,133,134,6975,135,136        <248>
9700 DATA M6,7110,A,B,C,7200,←,7300,N         <062>
9750 DATA M7,7600,133,7700,134                <189>
9900 DATA ENDE                                <113>
```

Listing »Sound-Editor« (Schluß)

Memory Map mit Wandervorschlägen (Teil 8)

Neben der dynamischen Tastaturabfrage werden die Adressen 153 bis 158 behandelt. Deren Aufgabe ist die Ausgabe von Fehlermeldungen.

Die dynamische Tastaturabfrage ist das Kernthema mit dem wir uns heute beschäftigen wollen. Zuvor aber noch einige Speicherstellen, die wie beim letzten Mal das Betriebssystem als Zeiger für Ein/Ausgabeoperationen benutzt.

Adresse 153 ($99)

Nummer des Eingabe-Gerätes

Das Betriebssystem verwendet diese Speicherzelle, um festzuhalten, welches Gerät zur Eingabe verwendet werden soll.

Die Nummern sind wie folgt festgelegt:

0	= Tastatur
1	= Datasette
2	= RS232 (User-)Port
3	= Bildschirm
4, 5	= Drucker
8-11	= Floppy-Laufwerke

Nach dem Einschalten oder nach RESET des Computers steht in 153 eine 0 (Tastatur). Nach jedem Einsatz eines anderen Gerätes wird diese Speicherzelle wieder auf 0 gesetzt, so daß wir immer die Tastatur zur Verfügung haben.

Für Maschinenprogrammierer ist diese Adresse sicherlich wertvoll. Die Routine, welche die Eingabegeräte festlegt, sobald der Befehl INPUT# beziehungsweise GET# ausgeführt wird, heißt CHKIN und beginnt beim C 64 ab Adresse 61966 ($F20E), beim VC 20 ab 62151 ($F2C7).

Für Basic-Programmierer habe ich in der Literatur nur eine Anwendung gefunden, und die wurde bereits bei der Besprechung der Speicherzelle 19 (Teil 3 des Kurses in Ausgabe 1/85, Seite 127) angekündigt.

Es ist dies eine MERGE-Routine. Leider funktioniert dieses Verfahren nicht bei dem 1541 Floppy-Laufwerk. Erfunden wurde die Routine von Brad Templeton und ist von Jim Butterfield unter dem Namen »Magic

Merge« für den VC 20/C 64 adaptiert worden. Ich gebe zu, in der Zwischenzeit sind noch andere, vielleicht auch kürzere MERGE-Routinen veröffentlicht worden. Aber diese hier verwendet gleich drei interessante Ingredienzen, nämlich die Speicherzellen 19 und 153, außerdem die sogenannte »Dynamische Tastenabfrage«. Wer die letztere nicht kennt, sollte sich zum Verständnis den Texteinschub 1 gleichen Namens ansehen.

Ein MERGE (deutsch: zusammenführen, verschmelzen) besteht darin, ein auf Band gespeichertes Programm zu einem im Computer stehenden anderen Programm so dazuzuladen, daß dieses nicht überschrieben, sondern ergänzt wird. Wichtig ist dabei, daß das Programm vom Band höhere Zeilennummern hat als das Programm im Computer. Außerdem muß das Programm auf dem Band als Datei gespeichert sein. Das wird so erreicht:

1. Programm eintippen
2. Direkt eingeben: OPEN 1,1,1,"Name":CMD1:LIST
3. Erst wenn READY kommt, direkt eingeben PRINT#1:CLOSE1

Damit ist das Programm auf dem Band gespeichert. Nun kommt der eigentliche MERGE-Vorgang.

4. Es steht ein Programm im Computer
5. Band mit dem Programm »Name« einlegen
6. Direkt eingeben: POKE 19,1:OPEN 1
7. Sobald READY erscheint, Bildschirm löschen (SHIFT-CLR).
8. Dreimal Cursor-Down
9. Direkt eingeben: PRINT CHR$(19):POKE 198,1: POKE 631,13:POKE 153,1
10. Das Band beendet den Ladevorgang mit einer Fehlermeldung, die wir ignorieren.
11. Nach CLOSE 1 sind beide Programme zusammengefügt.

Wie gesagt, Schritt 6 verwendet Zelle 19 (bitte dort nachlesen), Schritte 8 und 9 sind die »Dynamische Tastenabfrage«, und Schritt 9 verwendet zusätzlich die hier zur Diskussion stehende Speicherzelle 153, um die Datasette als Eingabegerät zu definieren.

Adresse 154 ($9A)

Nummer des Ausgabe-Gerätes

Diese Speicherzelle entspricht der Zelle 153, nur steht hier die Nummer des Gerätes, über das die Ausgabe läuft.

Nach dem Einschalten und nach Ausgabeoperationen wird der Wert immer auf 3 gesetzt. Das ist entsprechend der oben genannten Zuordnung der Bildschirm.

Für Maschinenprogrammierer sei erwähnt, daß Basic bei den Befehlen PRINT# oder CMD die Routine CHKOUT einsetzt, welche die Adresse 154 belegt. Sie steht im C 64 ab Adresse 62032 ($F250), im VC 20 ab 62217 ($F309).

Adresse 155 ($9B)

Fehlerkontrolle bei Bandoperationen

Die Commodore-Datasette ist deswegen so zuverlässig, weil sie mehrere Methoden zur Fehlererkennung beziehungsweise Korrektur von Lese- und Schreibfehlern verwendet.

Eine der Methoden ist die sogenannte Parity-Prüfung. Sie ist nichts anderes als eine Quersummenbildung der einzelnen Stellen jedes Bytes, deren Resultat überprüft wird.

Die Speicherzelle 155 wird bei dieser Parity-Prüfung eingesetzt.

Adresse 156 ($9C)

Flagge für korrektes Byte vom Band

In dieser Speicherzelle wird zwischengespeichert, ob das vom Band gelesene Byte die Prüfungen bestanden hat, also richtig ist oder nicht.

Adresse 157 ($9D)

Flagge für Meldungen

Man muß zwischen zwei Arten von Meldungen unterscheiden:
— Meldungen des Betriebssystems
— Meldungen des Basic-Übersetzers

Die Meldungen des Betriebssystems kennen wir als Angaben zum Ablauf, wie SEARCHING FOR, FOUND, PRESS PLAY ON TAPE und so weiter. Normalerweise nicht bekannt ist die Meldung I/O ERROR #, wobei nach dem Zeichen # Zahlen von 0 bis 29 stehen können. Diese Zahlen beziehen sich auf Meldungen des Übersetzers (Interpreter), die ausschließlich Fehlermeldungen sind. Das mag verwirrend klingen, klärt sich aber sofort. Die Flagge in 157 kann vier Werte annehmen, 0, 64, 128 und 192.

1. Der Wert 0 unterdrückt alle Meldungen des Betriebssystems. Dieser Modus tritt nach RUN beim Ablauf eines Programms ein.
2. Der Wert 64 läßt nur Fehlermeldungen des Betriebssystems zu. Dieser Modus ist normalerweise nicht vorgesehen, kann aber künstlich erzeugt werden.
3. Der Wert 128 unterdrückt die Fehlermeldung des Betriebssystems. Dieser Modus entspricht dem Normalfall.
4. Der Wert 192 läßt alle Meldungen zu. Auch dieser Modus ist nur künstlich herzustellen.

Das folgende Beispiel macht das deutlich. Geben Sie direkt ein:
POKE 157,0:LOAD "$",9

Wir versuchen, vom Gerät mit der Nummer 9, das ist eine zweite Floppy, die Directory zu laden. Wir erhalten entsprechend Punkt 1 nur die Meldung des Übersetzers »? DEVICE NOT PRESENT«. Verändern wir den POKE-Befehl für Punkt 2:
POKE 157,64:LOAD"$",9

Wir erhalten jetzt »I/O ERROR #5 ? DEVICE NOT PRESENT«.
POKE 157,128:LOAD"$',9
ergibt die Meldung »SEARCHING FOR $
? DEVICE NOT PRESENT«.

Schließlich nehmen wir noch den letzten Fall:
POKE 157,192:LOAD"$',9

Jetzt erhalten wir alles:
»SEARCHING FOR $
I/O ERROR #5
? DEVICE NOT PRESENT«

Da die Fehlermeldung des Betriebssystems und die zugehörigen Nummern in keinem Handbuch erwähnt sind, habe ich sie interessehalber in der folgenden Tabelle zusammengefaßt.

#	MELDUNG (ERROR)
1	TOO MANY FILES
2	FILE OPEN
3	FILE NOT OPEN
4	FILE NOT FOUND
5	DEVICE NOT PRESENT
6	NOT INPUT FILE
7	NOT OUTPUT FILE
8	MISSING FILE NAME
9	ILLEGAL DEVICE NUMBER
10	NEXT WITHOUT FOR
11	SYNTAX
12	RETURN WITHOUT GOSUB
13	OUT OF DATA
14	ILLEGAL QUANTITY
15	OVERFLOW
16	OUT OF MEMORY
17	UNDEF'D STATEMENT
18	BAD SUBSCRIPT
19	REDIM'D ARRAY
20	DIVISION BY ZERO
21	ILLEGAL DIRECT
22	TYPE MISMATCH
23	STRING TOO LONG
24	FILE DATA
25	FORMULA TOO COMPLEX
26	CAN'T CONTINUE
27	UNDEF'D FUNCTION
28	VERIFY
29	LOAD

Das nächste Mal fahren wir ab Speicherzelle 158 fort.

(Dr. Helmuth Hauck/ah)

Texteinschub 1. Dynamische Tastenabfrage

Jedesmal, wenn Sie eine Taste drücken, wird der ASCII-Codewert des Zeichens oder der Funktion dieser Taste ermittelt und im »Tastaturpuffer« gespeichert. Dieser Pufferspeicher liegt in den Speicherzellen 631 bis 640.

Die eigentliche Abfrage und Umcodierung eines Tastendrucks habe ich im Kurs »Alle Tasten-, Zeichen- und Steuercodes« in den Ausgaben 4/84 bis 7/84 beschrieben. Die ASCII-Codetabelle finden Sie da auch. Sie ist natürlich in allen Handbüchern enthalten, leider aber nicht immer ganz vollständig. Zuerst will ich Ihnen die Arbeitsweise des Tastaturpuffers zeigen.

Der Computer holt sich den ASCII-Codewert aus dem Tastaturpuffer und wenn gerade kein Programm läuft, druckt er das Zeichen auf den Bildschirm oder führt die Funktion der Taste aus. Das ist der oft zitierte »Direkt-Modus«.

Wenn aber ein Programm läuft, dann bleiben die Codezahlen im Puffer so lange stehen, bis der Computer fertig ist. Dann erst werden sie herausgeholt und verarbeitet. Das will ich Ihnen beweisen.

Tippen Sie im Direkt-Modus ein:
FOR K=1 TO 15000:NEXT K (RETURN)

Während diese an sich sinnlose Zeitschleife 15000mal im Kreise rumrennt, haben Sie genügend Zeit, mehrere Tasten zu drücken, zum Beispiel die erste Buchstabenreihe (QWERTYUIOP@*↑). Natürlich sehen Sie am Bildschirm gar nichts, denn das Programm der Schleife läuft ja noch. Sobald aber die Schleife zu Ende ist, erscheinen 10 der getippten Buchstaben. Quod erat demonstrandum!

Warum nur 10 Buchstaben? Nun, der Tastaturpuffer hat nur 10 Plätze, logisch?

Jetzt müssen wir noch eine weitere Speicherzelle ins Spiel bringen. In die Zelle 198 kann man nämlich eine Zahl hineinPOKEn, welche die Anzahl der Zeichen im Tastaturpuffer begrenzt.

Wiederholen Sie bitte das Experiment von oben, nur soll die direkt eingegebene Zeile erweitert werden:
FOR K=1 TO 15000:NEXT K: POKE 198,6 (RETURN)

Und siehe da, jetzt werden nur die 6 Buchstaben Q bis Y ausgedruckt. Diese Anwendung des Tastaturpuffers nutzen wir für das Kochrezept »**Dynamische Tastenabfrage**« aus. Allerdings müssen wir dazu prinzipiell die ASCII-Codewerte verwenden, so wie im nächsten Beispiel.

Löschen Sie bitte den Bildschirm und geben Sie ein (identisch für VC 20 und C 64):

```
10 PRINT CHR$(65)
20 PRINT CHR$(156)
30 PRINT CHR$(66)CHR$(13)CHR$(67)
```

65 ist der Code für A, 156 für die Farbe »purple«, 66 für B, 13 für »RETURN« und 67 für C. Das heißt also, daß wir zuerst ein A drucken, dann auf »purple« umschalten, darunter dann das B schreiben, einmal RETURN geben und dann noch das C folgen lassen.

Bild 1 zeigt den Ausdruck auf dem Bildschirm, wenn Sie diese Zeilen LISTen und starten.

Zur Erklärung: Die Leerzeile zwischen A und B ist bedingt durch die PRINT-Anweisung in Zeile 20, welche nur die Farbe umschaltet. Obwohl die Codes für B und C zusammen in einer Zeile stehen, werden sie doch durch das »RETURN« (13) untereinandergesetzt.

Anstelle der 13 können Sie alle mögliche anderen Steuerfunktionen setzen. Bild 2 zeigt das Resultat von 17, nämlich »Cursor down«. Wenn Sie die 8 nehmen, können Sie den Zeichensatz nicht mehr ändern. Der Einsatz der gleichzeitig gedrückten SHIFT- und Commodore-Taste funktioniert erst nach CHR$(9) wieder.

Als nächstes wollen wir die ASCII-Codewerte und den Tastaturpuffer im »Programm-Modus« einsetzen.

Das Resultat von Bild 2 wollen wir jetzt durch POKEn der ASCII-Werte in den Tastaturpuffer wiederholen.

```
5    POKE 198,5
10   POKE 631,65
20   POKE 632,156
30   POKE 633,66: POKE 634,17: POKE 635,67
99   END
```

Prinzipiell macht dieses Programm das gleiche wie das Programm in Bild 2. Trotzdem erhalten wir nach LIST und RUN ein anders Ergebnis, nämlich das von Bild 3.

Ist das ein Fehler? Natürlich nicht. Nach RUN laufen zuerst mal alle POKE-Befehle ab. Zeile 5 gibt an, wieviele Zeichen im Puffer stehen sollen. In Zeile 99 findet das Programm das ENDe und meldet sich mit READY. Jetzt erst wird im Tastaturpuffer nachgeschaut. Dort findet der Computer zuerst das A, dann »purpur«, dann das B welches sofort neben das A gesetzt wird. Das ist auch logisch, denn es fehlt ja jede Angabe, eine Zeile tiefer zu gehen. Um das zu erreichen, müssen wir in der Zeile 10 den Codewert für RETURN einschieben:

10 POKE 631,65: POKE 632,13

Vorsicht! Sie müssen in Zeile 20 und 30 alle POKE-Adressen um 1 erhöhen und auch die Zahl in Zeile 5. Nehmen sie 10, dann haben Sie Platz für Erweiterungen. So, jetzt LIST und RUN eingeben und es erscheint Bild 4 — und wir haben schon wieder ein Problem! Aber alles im Computer ist logisch! Nach dem A findet er den Wert für »RETURN« also führt er den Befehl aus, auf dem der Cursor gerade steht. Der steht auf dem A. Da das kein gültiger Basic-Befehl ist, druckt der Computer die Fehler-Meldung und zeigt READY an. Danach allerdings macht er weiter wie oben.


```
LIST
10 PRINT CHR$(65)
20 PRINT CHR$(156)
30 PRINT CHR$(66) CHR$
(13) CHR$(67)

READY.

RUN
A

B
C
READY.
```

Bild 1. Testprogramm

```
LIST
10 PRINT CHR$(65)
20 PRINT CHR$(156)
30 PRINT CHR$(66) CHR$
(17) CHR$(67)

READY.

RUN
A

B
 C
READY.
```

Bild 2. CR ersetzt durch »Cursor down«

```
LIST
5 POKE 198,5
10 POKE 631,65
20 POKE 632,156
30 POKE 633,66:POKE 63
4,17:POKE 635,67
99 END

READY.

RUN
READY.
AB
  C
```

Bild 3. Programm (Bild 2) in den Tastaturpuffer gePOKEt

Und jetzt kommt die ASCII-Zahl 141 (SHIFT-RETURN) voll zur Geltung. Diese Kombination nämlich setzt den Cursor an den Anfang der nächsten Zeile, ohne den Befehl »RETURN« auszuführen.

Ersetzen Sie also die 13 in Zeile 10 durch 141, dann läuft's (Bild 5).

Es gibt übrigens noch eine interessante ASCII-Codezahl, die in keiner Tabelle steht, nämlich 131. Sie bedeutet dasselbe wie die geSHIFTete STOP-Taste, also die Funktion »LOAD + RUN«.

Wenn Sie diesen Code mit PRINT CHR$(131) ausprobieren, funktioniert er allerdings nicht. Deshalb steht er wohl auch nicht in den Tabellen.

In den Tastaturpuffer gePOKEt bringt er aber seine Wirkung.

Setzen Sie bitte in Zeile 30 an die Stelle von 67 die Zahl 131 und anstelle des C erscheint

LOAD und PRESS PLAY ON TAPE. Gut, nicht wahr?

So, jetzt haben wir alle Zutaten für unser Kochrezept zusammen. Löschen Sie bitte alles bisherige und tippen Sie ein:

```
10   PRINT CHR$(147)
20   FOR I=1 TO 5: PRINT I
30   FOR T=1 TO 500: NEXT T
40   NEXT I
```

Nach Löschen des Bildschirms (Zeile 10) drucken wir zum Ausschmücken die Zahlen 1 bis 5 untereinander (Zeile 20 und 40) und damit es nicht zu schnell geht, bremsen wir mit der Zeile 30.

```
50   PRINT "LIST" (Das ist natürlich sehr einfach, aber jetzt kommt's!)
60   POKE 198,5
70   POKE 631,145: POKE 632, 145: POKE 633,145
80   POKE 634,13
90   END
```

Nach RUN erscheinen erst die fünf Zahlen, dann wird in einer neuen Zeile das Wort LIST geschrieben. Nach END wird erst (wie immer) eine Zeile ausgelassen, dann READY gedruckt und schließlich springt der Cursor an den Anfang der Zeile darunter. Während der Cursor anfangen will, drei Zeilen unter dem Wort LIST zu blinken, findet der Computer im Tastaturpuffer 3 den ASCII-Code für »Cursor up«. Also geht dieser auch drei Zeilen hoch und will jetzt auf dem Wort LIST blinken.

Damit wird es aber wieder nichts, denn im Puffer steht ja noch der Code für RETURN (13). Das wird ausgeführt und zwar für das LIST. Es hat dieselbe Wirkung, als ob Sie direkt LIST tippen und danach auf die RETURN-Taste drücken, nämlich das Programm wird ausgeLISTet.

Alle sinnvollen Basic-Befehle, die Sie in der Zeile 50 PRINTen, werden durch diese dynamische Manipulation des Cursors ausgeführt.

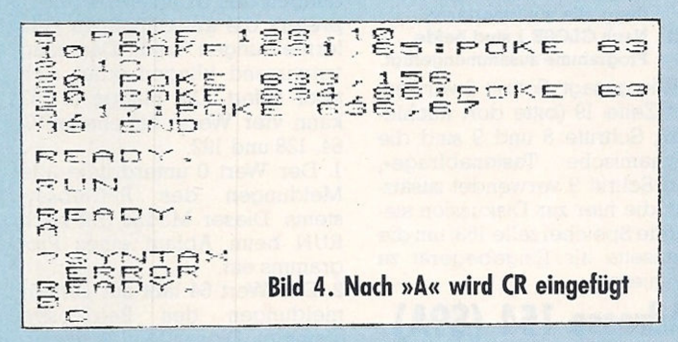
Versuchen Sie es mit

```
50   PRINT "PRINT 16 * 35"
50   PRINT "LOAD"
50   PRINT "GOTO 10"
50   PRINT "RUN"
50   PRINT "RUN 50"
```

und falls Sie dieses kleine Programm geSAVEt haben

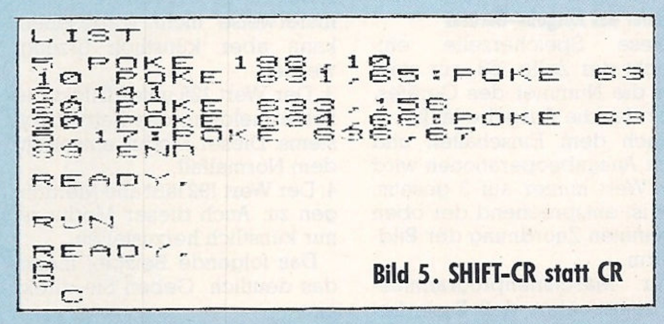
```
50   PRINT "SYS 64824"
50   PRINT "SYS 64763"
```

Die Kunst ist also, mit entsprechenden Codezahlen den Cursor an diejenige Stelle des Bildschirms zu bringen, wo innerhalb eines Programms eine geeignete Anweisung gedruckt worden ist. Man kann damit getrennte Programmteile nachladen, mit SYS-Befehlen Maschinenprogramme aufrufen, oder gar Programme durch sich selbst ändern lassen.



```
5 POKE 198,10
10 POKE 631,65:POKE 63
2,13
20 POKE 633,156
30 POKE 634,66:POKE 63
5,17:POKE 636,67
99 END
READY.
RUN
READY.
A
?SYNTAX
  ERROR
READY.
B
 C
```

Bild 4. Nach »A« wird CR eingefügt



```
LIST
5 POKE 198,10
10 POKE 631,65:POKE 63
2,141
20 POKE 633,156
30 POKE 634,66:POKE 63
5,17:POKE 636,67
99 END
READY.
RUN
READY.
A
B
 C
```

Bild 5. SHIFT-CR statt CR

Logeleien (Teil 1)

In vielen Basic-Programmen tauchen sie auf, die logischen Operatoren wie NOT, AND und OR. Bei sinnvollem Einsatz kann man eine Menge Basic-Speicherplatz sparen. Aber auch Ausdrücke wie IF A THEN ... bringen nicht nur den Anfänger manchmal ins Schleudern. Was steckt eigentlich im Endefekt dahinter?

Das hätte sich der »Vater der Logik«, Aristoteles, vor 2200 Jahren nicht träumen lassen können, was aus seinen Gedankengängen einmal werden würde. Man kann aber mit einiger Berechtigung behaupten, daß es ohne seine Werke und die von Leibniz (vor etwa 300 Jahren hat dieser die Logik modernisiert) keine Computer gäbe. Die Arbeiten der Begründer der mathematischen Logik, Boole (1815-64) und Frege (1848-1925), basieren auf dem jahrtausende alten Fundament. Wie für den Computer geschaffen — aber eigentlich ist es umgekehrt: Die logische Maschine ist ein Produkt der mathematischen Logik — ist die sogenannte zweiwertige Logik. Hier geht es um »Wahr« und »Falsch«, ein drittes gibt es nicht (tertium non datur, wie die alten Philosophen zu sagen pflegten). Die Verwandtschaft liegt auf der Hand: Die elektronische Maschine kennt die Zustände Strom an/Strom aus oder Ladung vorhanden/nicht vorhanden oder eben einfach 1 und 0, die Ziffern der Binärarithmetik.

Dazu noch eine technische Bemerkung: Sollten Sie mit dem System der binären Zahlen noch nicht vertraut sein, dann lesen Sie den beigefügten Kasten, in dem die wichtigsten Begriffe erklärt sind.

1. Wahrheitswerte und Aussagen

Was sind Aussagen? Am besten sehen wir uns einfach einige Beispiele an:

— Rosen sind rot
— Die Zahl ist größer als 100.

Das sind einfache Aussagen. Eine zusammengesetzte Aussage ist beispielsweise:

Rosen sind rot **UND** Veilchen sind blau. Hier wurden zwei Aussagen UND-verknüpft. Keine Aussage ist:

— Wohin gehst Du?

Ich hoffe, daß auf diese Weise ohne tiefschürfende philosophische Erörterungen (denn die gibt es natürlich) das Wesen von Aussagen klar geworden ist.

Aussagen können wahr sein. Sie enthalten dann den Wahrheitswert »Wahr«. Sie können aber auch statt dessen falsch sein, was ihnen den Wahrheitswert »Falsch« verleiht. Der Wahrheitswert von zusammengesetzten Aussagen hängt zum einen von den Wahrheitswerten der Einzelaussagen ab und zum anderen von der Art ihrer Verknüpfung. Wir werden im folgenden für die Wahrheitswerte die Buchstaben W (für Wahr) und F (für Falsch) verwenden.

In der binären Arithmetik entspricht die 1 dem W und die 0 dem F. Keine Angst, falls Ihnen das noch unverständlich erscheint, wir werden noch allerlei Logeleien mit Binärzahlen treiben. Dann wird Ihnen das ganz geläufig sein. Was uns momentan interessiert, ist, wie der Computer Wahrheitswerte angibt. Dazu machen wir einen kleinen Test mit dem folgenden Programm:

```
10  INPUTA,B
20  C=(A=B)
30  PRINTA,B,C
```

In Zeile 20 ist eine Aussage enthalten: A=B. Wenn Sie A und B so eingegeben haben, daß beide gleich sind, dann ist diese Aussage wahr (W), und in C finden Sie den Wahrheitswert, den der Computer für W verwendet. Probieren Sie dies mal aus: Sie finden —1! Wenn aber A und B verschiedene Zahlen waren, ist die Aussage A=B falsch, und in C steht 0, was die Computerform von F ist. Weshalb —1 als Wahrheitswert des Commodore 64 für W? Das liegt daran, daß unser Computer für den Wahrheitswert ein ganzes Byte reserviert. Das schreibt er im Falle W voller Einsen:

1111 1111

Nach den Regeln von vorzeichenbehafteten Binärzahlen ist das aber die Erscheinungsform von —1.

2. Das merkwürdige IF A THEN ...

Bei IF..THEN-Verzweigungen wird vom Computer der gesamte Ausdruck zwischen IF und THEN auf seinen Wahrheitswert untersucht (das kann dann eine Aussage sein, aber auch etwas anderes, zum Beispiel eine Variable). Dabei nimmt der C 64 alles für wahr an, was bei dieser Untersuchung ungleich Null ist. Wenn also in einer Verzweigung der Form:

IF A THEN ...

A schon irgendeinen Wert angenommen hat, der ungleich Null ist, wird die Bedingung als erfüllt angesehen. Probieren Sie mal mit verschiedenen Eingaben für A:

```
10  INPUT A
20  IF A THEN PRINT "TEST"
30  PRINT A
```

In der uns geläufigeren Form dieser Verzweigung tauchen zwischen IF und THEN Aussagen in mathematischer Form auf, in denen die Vergleichssymbole verwendet werden, zum Beispiel:

IF A >= 100 THEN GOTO ...

3. Rechnen mit Wahrheitswerten

Außerhalb solch einer Verzweigung ist es manchmal gar nicht so einfach festzustellen, ob es sich bei einem Ausdruck um eine Aussage dreht oder vielleicht nur um eine Zuweisung. Es kommt hier sehr auf den Zusammenhang mit dem übrigen Programmtext an. Beispielsweise kann A=B sowohl eine Zuweisung sein (LET A=B) als auch eine Aussage (C=(A=B)). In Programmen, in denen viel mit logischen Operatoren gearbeitet wird, empfiehlt sich deshalb — um mehr Klarheit zu schaffen —, die vom C 64 verstandene, aber so gut wie nie verwendete Basic-Vokabel LET bei Zuweisungen zu verwenden.

Ein noch eklatanteres Beispiel ist enthalten in A=5+2. Das ist offensichtlich keine Aussage, könnte deutlicher aber noch als LET A=5+2 geschrieben werden. In einem anderen Zusammenhang ist es eine Aussage:

C=4+(A=5+2)

Ist hier A=5+2 wahr (wenn A=7), ergibt sich ein C von 3, weil ja der Wahrheitswert der Aussage —1 beträgt. Im anderen Fall ist C=4. Man kann mit Aussagen — besser gesagt, mit ihren Wahrheitswerten — also auch rechnen.

Im folgenden Programmbeispiel sind gleich fünf Wahrheitswerte miteinander rechnerisch verknüpft. Es handelt sich um eine kurze Routine zur Umrechnung von ASCII-Code in Bildschirmcode:

```
10  INPUT"ASCII-WERT";A
20  C=A-161-33*(A<255)-64*(A<192)-32*(A<160)+32*(A<96)-64*(A<64)
30  PRINT"ASCII="A,"POKE-CODE="C
```

Beispiel: ASCII=65, Bildschirmcode=1

Aussagen können auch mit Strings gebildet werden. Auch dabei ergeben sich die Wahrheitswerte —1 oder 0. Allerdings lassen sich hier sinnvoll nur die Vergleichsrelationen »=« und »ungleich« einsetzen. Alle anderen »größer als«, »kleiner als« etc. sind mit Vorsicht zu genießen. Zur Ermittlung des Wahrheitswertes werden nämlich von links nach rechts alle Buchstaben beider Strings in Form ihrer ASCII-Werte miteinander verglichen. Die erste auf diese Weise gefundene Abweichung führt dann zum Ergebnis. Ein Beispiel soll das Problem verdeutlichen:

Die Aussage (A$<B$) soll uns einen Wahrheitswert liefern. Wenn nun A$="DZZZ" und B$="EAAA" definiert wurden, ergibt sich —1 als Wahrheitswert, weil der ASCII-Wert von D kleiner ist als der von E. Der Rest der Strings spielt keine Rolle mehr.

4. Die logischen Operatoren in Basic

Drei Operatoren haben wir im Normalfall zur Verfügung für die sogenannte Boolèsche Algebra, wie man manchmal auch das Rechnen mit logischen Operatoren nennt:

NOT, AND, OR

Die Tabelle 1 zeigt, wo in der Prioritätenreihenfolge aller Operatoren diese drei einzuordnen sind. Ebenso wie bei den arithmetischen Operatoren (+,*,↑ etc.) kann auch hier die Reihenfolge durch sinnvolle Anwendung von Klammern verändert werden.

Bemerkung	Operatoren	Beispiel
Höchste	↑	A↑B
Priorität	*, /	A*B, A/B
	+, —	A+B, A-B
	NOT	NOT A
	AND	A AND B
Niedrigste		
Priorität	OR	A OR B

Tabelle 1. Reihenfolge der Priorität von Operatoren

5. Wahrheitstabellen

Jede Verkoppelung von Aussagen (beziehungsweise ihren Wahrheitswerten) mittels logischer Operatoren ergibt ebenfalls wieder Wahrheitswerte. Wir hatten vorhin schon er-

wähnt, daß das Ergebnis abhängt:
a) von den Wahrheitswerten der zu verknüpfenden Aussagen und
b) von der Art der Verknüpfung.

Das kann manchmal schon reichlich schwer zu durchschauen sein, was welche Kombination ergibt. Deshalb schreibt man sich eine Übersicht, die alle möglichen Ausgangs-Wahrheitswerte enthält und die nach der Verknüpfung entstandenen Wahrheitswerte. Solche Tabellen nennt man Wahrheitstabellen. Wir werden diese Tabellen im folgenden bei der Erklärung der logischen Operatoren verwenden. Wie man sie erstellt und liest, wird Ihnen schnell geläufig sein.

Noch eine Bemerkung: Die Anwendung der logischen Operatoren auf Aussagen und auf Zahlen wird bei uns — aus Gründen der besseren Verständlichkeit — getrennt behandelt werden. Das ist eigentlich nicht nötig, denn im Grunde genommen sind ja die Wahrheitswerte von Aussagen auch nur Zahlen.

6. Die Anwendung von NOT auf Aussagen

Der NOT-Operator erzeugt immer das Gegenteil des Wahrheitswertes der Ausgangsaussage. Wenden wir also NOT auf eine Aussage mit dem Wahrheitswert W an, dann ergibt das den Wert F. Dasselbe geschieht im umgekehrten Fall.

Dazu wollen wir uns eine Wahrheitstabelle schreiben (Tabelle 2). In der linken Spalte sind die beiden möglichen Wahrheitswerte einer Aussage A aufgeführt. Die rechte Spalte ergibt sich durch Anwendung von NOT auf die Werte der linken Spalte:
NOT W = F
NOT F = W

A	NOT A
W	F
F	W

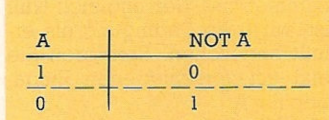
Tabelle 2. Wahrheitstabelle: NOT bei Aussagen

Das ist gar nicht so schwer, nicht wahr? An einem Beispiel wollen wir uns das auch noch mal ansehen. Die Zeile:
10 A=(3=5):PRINT NOT A
ergibt den Wert —1. Die Aussage 3=5 ist ja falsch (F oder für den Computer 0) und A deshalb 0. NOT A ergibt dann das Gegenteil, also W oder —1.

Eine Basic-Zeile wie die folgende:
10 A = NOT A
kann man wie einen Flip-Flop-Schalter einsetzen, falls man den Wahrheitswert von A zuvor als 0 oder —1 definiert hat. Jedesmal, wenn das Programm über diesen Ausdruck läuft, kippt der Wert von A.

7. NOT auf Zahlen angewendet

Zahlen oder Variable, auf die der NOT-Operator angewandt wird, verkehren sich in ihr Einerkomplement. Falls Sie mit den Begriffen »Einerkomplement« und »Zweierkomplement« Schwierigkeiten haben, weise ich Sie nochmal auf den hier abgedruckten Kasten mit Erläuterungen hin. Jedes Bit wird also »gekippt«. Aus einer 0 wird 1 und umgekehrt. Die Wahrheitstabelle kann hier ähnlich aufgestellt werden wie für Aussagen, nur tauchen hier anstelle von W und F nun 1 und 0 (also die Bitwerte) auf (siehe Tabelle 3).



A	NOT A
1	0
0	1

Tabelle 3. Wahrheitstabelle: NOT bei Zahlen

Eine Anwendung von NOT wäre beispielsweise das gezielte Kippen von Bytes oder 2-Byte-Werten. Besonders bei Sprite-Programmen ist sowas denkbar. Nehmen wir mal an, wir hätten acht Sprites definiert, von denen vier bis zu einem bestimmten Ereignis sichtbar sein sollen. Die anderen vier sind solange abgeschaltet. Wenn das Ereignis (Tastendruck oder Zusammenstoß ...) eintritt, werden die obersten vier ab- und gleichzeitig die anderen vier angeschaltet. Register 21 (53269) des VIC-II-Chip stellt für jedes Sprite ein Schalt-Bit zur Verfügung (0 steht dort für ein ausgeschaltetes und 1 für ein aktiviertes Sprite). Sollen also zuerst die Sprites 0, 2, 4, 6 an- und 1, 3, 5, 7 abgeschaltet sein, dann enthält Register 21 das Bit-Muster:
0101 0101

Darauf nun eine NOT-Operation angewendet ergäbe genau das erwünschte Kippen:
1010 1010 nach NOT.

Es läge daher nahe, etwa so zu programmieren:
POKE 53269,NOT PEEK(53269)

Leider ist das nicht möglich. Versucht man es trotzdem, erhält man einen ILLEGAL QUANTITY ERROR. Woran liegt das? Die Ursache dafür liegt wieder in der Eigenart, wie der C 64 Integer-Zahlen interpretiert. Zwar erzeugt die NOT-Operation das Einerkomplement der Ausgangszahl, verstanden wird dieses Ergebnis aber als Zweierkomplementzahl. Daraus folgt, daß man mit der obigen Anweisung versucht, eine negative Zahl in die Speicherstelle zu POKEn, und da streikt unser Computer. Die Binärzahl 1010 1010 wird als —86 verstanden. Was kann man tun, daß der wahre Wert, also 170, in den Speicher gePOKEt wird? Bei 1-Byte-Zahlen ist das relativ einfach: Man wandelt sie durch Abziehen von 256 in eine 2-Byte-Zahl um, also NOT(A-256). In Bild 2 ist zu sehen, was dabei geschieht.

Ein anderer Weg für 1-Byte-Zahlen (also Zahlen, die kleiner als 255 sind) ist NOT A AND 255, was wir aber erst später verstehen werden, wenn wir die AND-Operation kennengelernt haben. Vorsicht muß man walten lassen, wenn auch negative A-Werte auftreten. Negative A können ohne Probleme der NOT-Operation unterzogen werden, weil das Ergebnis in jedem Fall positiv ist. Verwendet man NOT(A-256), kommt es zu Fehlern. Der Weg über NOT A AND 255 ist auch mit negativen 1-Byte-Zahlen möglich. Am sichersten wäre also diese Basic-Zeile:
POKE53269,NOTPEEK(53269) AND255

2-Byte-Zahlen spielen bei POKEs keine Rolle, weshalb wir auf die Umwandlung der negativen Zahl in den vorzeichenfreien Ausdruck hier nicht eingehen werden. Eine andere Sache muß bei der Anwendung von NOT noch bedacht werden: Diese Operation arbeitet mit Integer-Zahlen. Treten nun als Argumente »Komma-Zahlen« auf, dann führt der C 64 zuerst eine INT-Operation aus, die unter Umständen zu Fehlern führen kann: 3 999 ergibt dann ein deutlich anderes Ergebnis als 4 001.

Die NOT-Operation kann nur auf Zahlen, nicht aber auf Strings angewendet werden. Allerdings erfolgt die computerinterne Speicherung von Buchstaben ja als ASCII-Werte, also auch als Zahlen. Wenn wir daher die ASCII-Codes eines Textes der NOT-Verknüpfung unterziehen, haben wir auch Strings dieser Operation zugänglich gemacht. Genau das tut ein kleines Demo-Programm (Programm Logik-1): Es ver- und entschlüsselt Texte mittels NOT.

In diesem Demo-Programm wird Text Buchstabe für Buchstabe in die ASCII-Werte übertragen und diese dann durch NOT komplementiert und in C$ gespeichert. Der Ausdruck ist dann ein unverständliches Kauderwelsch, das aber durch eine zweite NOT-Operation wieder lesbar wird.

8. Eine kleine Hilfe

Als Programm Logik-2 finden Sie eine kleine Denk- und Rechenhilfe hier abgedruckt. Dieses Programm kann Zahlen zwischen —32767 und +32767 den logischen Verknüpfungen unterziehen und zeigt Ihnen dabei, was sich im Binärformat abspielt und wie die Argumente — falls negative Zahlen auftreten — im vorzeichenfreien Format aussehen. Nach dem RUN bietet Ihnen ein Menü fünf Möglichkeiten an: NOT, AND, OR, EOR und Programmende. NOT kennen Sie ja schon, die anderen logischen Operatoren werden Ihnen in der nächsten Folge geläufig werden. Dort wird Ihnen dieses Programm dann einige Hilfestellungen geben für etwas kompliziertere Sachverhalte. Wenn Sie eine Option angewählt haben, werden zwei Zahleneingaben verlangt (bei NOT nur eine) und dann die entsprechenden Binärzahlen, die vorzeichenfreien Formen und das Ergebnis der Verknüpfung ausgedruckt. Innerhalb der gesteckten Grenzen sind alle Zahleneingabe-Formate möglich. Ein Tastendruck bringt Sie wieder ins Menü zurück.

In der nächsten Folge werden wir uns die Operatoren AND und OR ansehen. Ein in dem WAIT-Befehl versteckter weiterer Operator EOR soll aufgespürt werden, und wir werden wieder einige Anwendungen der Boole'schen Algebra in unserem Computer kennenlernen.

(Heimo Ponnath/gk)

```
10 REM-------------------------------------      <102>
20 REM        LOGIK - 1                          <119>
30 REM-------------------------------------      <122>
40 :                                             <098>
50 PRINT"{CLR,7SPACE}GEHEIMCODE MIT NOT"         <071>
60 PRINT:PRINT                                   <168>
70 PRINT" GEBEN SIE EIN WORT EIN";               <164>
100 INPUT A$:L=LEN(A$):DIM A(L),B(L),B$(L)
   :K=0                                          <062>
110 C$=""                                        <203>
120 FOR I=1 TO L:A(I)=ASC(MID$(A$,I,1)):B(
   I)=NOT A(I)-257:B$(I)=CHR$(B(I)):NEXT
   I                                             <190>
130 FOR I=1 TO L:C$=C$+B$(I):NEXT I              <136>
140 IF K=0 THEN:PRINT:PRINT"TEXT{11SPACE}=
    ";A$                                         <247>
145 IF K=1 THEN:PRINT:PRINT"GEHEIMCODE{5SP
   ACE}= ";A$                                    <130>
150 K=K+1:IF K=1 THEN A$=C$:GOTO 110             <038>
160 PRINT:PRINT"ENTSCHLUESSELT = ";C$            <101>
```

Programm Logik-1. Mit NOT eine Geheimschrift erzeugen

```
1 REM *********************************       <250>
2 REM *                               *       <229>
3 REM *      LOGISCHE  OPERATIONEN    *       <120>
4 REM *                               *       <231>
5 REM * ZUM VERFOLGEN IM BINAERFORMAT *       <156>
6 REM *          NOT,AND,OR,EOR       *       <184>
7 REM *                               *       <234>
8 REM *  HEIMO PONNATH  HAMBURG 1985  *       <082>
9 REM *                               *       <236>
10 REM********************************        <003>
15 PRINT CHR$(147):GOTO 200                   <177>
20 REM ***** UP-CURSOR SETZEN *****           <208>
25 POKE 211,S:POKE 214,Z:SYS 58640:RETURN     <088>
30 REM ***** UP-EINGABETEST *******           <162>
35 E=ABS(E):A=(E>=0):B=(E<255)                <207>
40 IF A AND B THEN M=8:RETURN                 <139>
45 A=(E>=255):B=(E<32768)                     <054>
50 IF A AND B THEN M=16:RETURN                <196>
55 M=0:RETURN                                 <046>
60 REM ***** UP-EINGABE 1 *********           <005>
65 S=1:Z=5:GOSUB 25:PRINT CHR$(28)"WELCHE
   ZAHL SOLLS DENN SEIN?"                     <192>
70 Z=7:GOSUB 25:PRINT"SIE MUSS ZWISCHEN -3
   2767UND +32767 SEIN"                       <226>
75 Z=9:GOSUB 25:INPUT W:E=W:GOSUB 35:IF M=
   0 THEN POKE 781,9:SYS 59903:GOTO 75        <211>
80 PRINT CHR$(30):RETURN                      <044>
85 REM ***** UP-EINGABE 2 *********           <031>
90 S=1:Z=5:GOSUB 25:PRINT CHR$(28)"GEBEN S
   IE NUN DIE BEIDEN ZAHLEN EIN."             <074>
95 S=1:Z=7:GOSUB 25:PRINT"SIE MUESSEN ZW.
   -32767 U. +32767 SEIN!"                    <116>
100 Z=9:GOSUB 25:INPUT"ERSTE ZAHL =";W1:E=
    W1:GOSUB 35:M1=M                          <043>
105 IF M=0 THEN POKE 781,9:SYS 59903:GOTO
    100                                       <156>
110 Z=11:GOSUB 25:INPUT"ZWEITE ZAHL=";W2:E
    =W2:GOSUB 35                              <254>
115 IF M=0 THEN POKE 781,11:SYS 59903:GOTO
    110                                       <208>
120 M=16+8*((M1=8)AND(M=8)):PRINT CHR$(30)
    :RETURN                                   <195>
125 REM **** UP-UMRECHNUNG DEZ/BIN ****       <021>
130 BI$="":DI=DE                              <041>
135 DI=DI/2:D$="0":IF DI<>INT(DI)THEN D$="
    1"                                        <121>
140 DI=INT(DI):BI$=D$+BI$:IF DI>0 THEN 135    <045>
145 IF LEN(BI$)<M THEN BI$="0"+BI$:GOTO 14
    5                                         <017>
150 RETURN                                    <036>
155 REM **** UP-UMRECHNUNG BIN/DEZ ****       <051>
160 DE=0:FOR I=1 TO M:IF MID$(A$,I,1)="1"T
    HEN DE=DE+2↑(M-I)                         <168>
165 NEXT I:RETURN                             <056>
170 REM **** UP-BILDSCHIRMAUSGABE *****       <094>
175 PRINT CHR$(158)TAB(5)DE;TAB(32-M)BI$;C
    HR$(30):RETURN                            <025>
198 REM                                       <085>
199 REM **** HAUPTPROGRAMM-VARIABLE ***       <214>
200 S=0:Z=0:E=0:A=0:B=0:M=0:W=0:W1=0:W2=0:
    M1=0:I=0:K=0:X=0                          <142>
205 DI=0:DE=0:C=0:V=0                         <178>
210 BI$="":D$="":A$="":B$="":C$="":Z$="---
    -------------------":BE$=""               <087>
220 DIM DE(3),BI$(3)                          <206>
299 REM **** HAUPTPROGRAMM-MENUE ******       <237>
300 POKE 53280,0:POKE 53281,0:PRINT CHR$(1
    47)CHR$(30)                               <084>
305 Z=3:S=2:GOSUB 25:PRINT CHR$(18)" DIE L
    OGISCHEN BEFEHLE IN BINAERFORM "          <125>
310 Z=7:GOSUB 25:PRINT TAB(10)"NOT"TAB(25)
    "1"                                       <133>
315 Z=9:GOSUB 25:PRINT TAB(10)"AND"TAB(25)
    "2"                                       <111>
320 Z=11:GOSUB 25:PRINT TAB(10)"OR"TAB(25)
    "3"                                       <108>
325 Z=13:GOSUB 25:PRINT TAB(10)"EOR"TAB(25
    )"4"                                      <185>
330 Z=15:GOSUB 25:PRINT TAB(10)"PROGRAMMEN
    DE"TAB(25)"5"                             <092>
335 POKE 646,10:Z=20:GOSUB 25:PRINT"BITTE
    WAEHLEN SIE EINEN MENUEPUNKT..."CHR$(3
    0)                                        <222>
340 GET B$:IF B$<"1"OR B$>"5"THEN 340         <045>
345 ON VAL(B$)GOSUB 400,500,600,700,800       <146>
350 Z=23:S=1:GOSUB 25:PRINT CHR$(3)"WEITER
     DURCH TASTENDRUCK..."                    <232>
355 POKE 198,0:WAIT 198,1:GOTO 300            <026>
399 REM **** OPTION - NOT ***********         <140>
400 PRINT CHR$(147):Z=2:S=1:GOSUB 25:PRINT
     CHR$(18)" OPTION{2SPACE}NOT{6SPACE}"C
    HR$(146)                                  <009>
405 BI$="":BE$="":D$="":W=0:DE=0:K=0:GOSUB
     65:IF W<0 THEN 450                       <158>
410 DE=W:GOSUB 130                            <136>
412 Z=15:S=0:GOSUB 25:GOSUB 175:PRINT TAB(
    15)Z$+" NOT":BE$=""                       <077>
415 FOR I=1 TO M:D$=MID$(BI$,I,1):IF D$="1
    "THEN D$="0":GOTO 420                     <016>
417 D$="1"                                    <049>
420 BE$=BE$+D$:NEXT I:IF K=1 THEN RETURN      <236>
425 BI$=BE$:A$=BE$:GOSUB 160:Z=17:S=0:GOSU
    B 25:GOSUB 175:BE$="":RETURN              <098>
440 REM **** UP-NEGATIVE BINAERZAHL ***       <115>
450 W=NOT W:DE=W:GOSUB 130:K=1:GOSUB 415:A
    $=BE$:GOSUB 160:K=0:IF X=1 THEN RETURN    <177>
455 BI$=BE$:BE$="":GOTO 412                   <009>
499 REM **** OPTION - AND ***********         <210>
500 PRINT CHR$(147):Z=2:S=1:GOSUB 25:PRINT
     CHR$(18)" OPTION{2SPACE}AND{6SPACE}"C
    HR$(146)                                  <079>
502 REM **** UP-EINGABENVERARBEITUNG **       <027>
505 GOSUB 90:IF W1<0 THEN W=W1:X=1:GOSUB 4
    50:X=0:BI$(1)=BE$:DE(1)=DE:BE$="":DE=0    <105>
510 IF W2<0 THEN W=W2:X=1:GOSUB 450:X=0:BI
    $(2)=BE$:DE(2)=DE:BE$="":DE=0             <066>
515 IF W1>=0 THEN DE=W1:GOSUB 130:DE(1)=DE
    :BI$(1)=BI$:DE=0:BI$=""                   <064>
520 IF W2>=0 THEN DE=W2:GOSUB 130:DE(2)=DE
    :BI$(2)=BI$:DE=0:BI$=""                   <073>
525 Z=15:S=0:GOSUB 25:DE=DE(1):BI$=BI$(1):
    GOSUB 175                                 <206>
530 Z=16:GOSUB 25:DE=DE(2):BI$=BI$(2):GOSU
    B 175:IF V=1 THEN RETURN                  <154>
532 PRINT TAB(15)Z$+" AND":BI$(3)=""          <131>
535 FOR I=1 TO M:A=VAL(MID$(BI$(1),I,1)):B
    =VAL(MID$(BI$(2),I,1)):C=A AND B          <220>
540 BI$(3)=BI$(3)+RIGHT$(STR$(C),1):NEXT I
    :A$=BI$(3):GOSUB 160:BI$=BI$(3)           <100>
545 Z=18:GOSUB 25:GOSUB 175:RETURN            <242>
599 REM **** OPTION - OR ************         <047>
600 PRINT CHR$(147):Z=2:S=1:GOSUB 25:PRINT
     CHR$(18)" OPTION{3SPACE}OR{6SPACE}"CH
    R$(146)                                   <130>
605 V=1:GOSUB 505:V=0                         <107>
610 PRINT TAB(15)Z$+" OR":BI$(3)=""           <159>
615 FOR I=1 TO M:A=VAL(MID$(BI$(1),I,1)):B
    =VAL(MID$(BI$(2),I,1)):C=A OR B           <045>
620 BI$(3)=BI$(3)+RIGHT$(STR$(C),1):NEXT I
    :A$=BI$(3):GOSUB 160:BI$=BI$(3)           <180>
625 Z=18:GOSUB 25:GOSUB 175:RETURN            <066>
699 REM **** OPTION - EOR ***********         <174>
700 PRINT CHR$(147):Z=2:S=1:GOSUB 25:PRINT
     CHR$(18)" OPTION{2SPACE}EOR{6SPACE}"C
    HR$(146)                                  <043>
705 V=1:GOSUB 505:V=0                         <207>
710 PRINT TAB(15)Z$+" EOR":BI$(3)=""          <072>
715 FOR I=1 TO M:A=VAL(MID$(BI$(1),I,1)):B
    =VAL(MID$(BI$(2),I,1))                    <047>
720 IF A+B=1 THEN C=1:GOTO 730                <151>
725 C=0                                       <252>
730 BI$(3)=BI$(3)+RIGHT$(STR$(C),1):NEXT I
    :A$=BI$(3):GOSUB 160:BI$=BI$(3)           <034>
735 Z=18:GOSUB 25:GOSUB 175:RETURN            <176>
799 REM **** OPTION-PROGRAMMENDE ****         <136>
800 PRINT CHR$(147):S=8:Z=12:GOSUB 25:PRIN
    T"DAS WAR'S...TSCHUESS !"                 <014>
805 Z=22:S=0:GOSUB 25:END                     <239>
```

Programm Logik-2. Sie sehen, was bei der Verknüpfung von Zahlen mit logischen Operatoren geschieht. Beachten Sie bitte die Eingabehinweise auf Seite 77

Einige Erläuterungen zum Rechnen mit Binärzahlen

a) Dezimales und binäres Zahlensystem

Wir sind es gewohnt, im dezimalen Zahlensystem zu rechnen unter Verwendung der Ziffern 0 bis 9. Die Grundlage unseres Zahlensystems ist die Zahl 10. Reichen die Ziffern 0 bis 9 nicht aus, um eine Zahl (zum Beispiel 9+1) darzustellen, dann führen wir den Stellenwert einer Zahl ein. Hierbei hat jede Stelle in einer Zahl einen Faktor, der sich berechnet aus 10^{Stelle}. Zum Beispiel in der Zahl 132 befindet sich die Ziffer 2 an der nullten Stelle, und sie wird mit dem Faktor 10^0 multipliziert ($10^0=1$). Für unsere Beispielszahl können wir so schreiben:

132=1*10↑2+3*10↑1+2*10↑0 =1*100+3*10+2*1

Nach demselben Prinzip wird im Binärsystem verfahren. Hier gibt es allerdings nur die Ziffern 0 und 1. Die Grundlage dieses Systems ist die Zahl 2. Auch hier spielt der Stellenwert die entscheidende Rolle. Der Faktor ist nun aber die entsprechende Zweierpotenz (2↑Stelle). Die Binärzahl 110 beispielsweise setzt sich dann so zusammen:

110=1*2↑2+1*2↑1+0*2↑0
=1*4+1*2+0*1=6

b) Umrechnung Binär- in Dezimalzahl

Damit haben Sie praktisch schon gesehen, wie man Binärzahlen in Dezimalzahlen umrechnen kann: Jeder Stelle entspricht eine Zweierpotenz. Alle Zweierpotenzen, an deren Stelle eine 1 steht, werden zusammengezählt. Dabei gilt: $2^0=1$ (siehe Bild 1).

4. Stelle	3. Stelle	2. Stelle	1. Stelle	0. Stelle
$=1*2^4$	$+1*2^3$	$+1*2^2$	$+0*2^1$	$+1*\ 2^0 =$
= 1*16	+1*8	+1*4	+0*2	+1* 1 =
= 16	+ 8	+ 4	+	1 =
= 29				

Bild 1. Umrechnung einer Binärzahl (11101) in eine Dezimalzahl (29)

c) Umrechnung Dezimal- in Binärzahl

Der umgekehrte Weg läßt sich durch eine Kettendivision der Dezimalzahl durch 2 realisieren. Jeweils die Divisionsreste ergeben eine Stelle der Binärzahl. Beispiel (die Zahl 29):

		Binärzahl
29: 2 = 14	Rest 1	0. Stelle
14: 2 = 7	Rest 0	1. Stelle
7: 2 = 3	Rest 1	2. Stelle
3: 2 = 1	Rest 1	3. Stelle
1: 2 = 0	Rest 1	4. Stelle

Bild 1a. Umrechnung einer Dezimalzahl in eine Binärzahl

Die Kettendivision ist beendet, wenn sich als Ergebnis 0 zeigt. Die Zahl 29 lautet also im Binärformat 11101.

d) Bits und Bytes

Eine Stelle einer solchen Binärzahl nennt man ein »Bit«. Unser Computer ist so gebaut, daß er 8 Bit zu einer Einheit zusammenfaßt. Solch eine Einheit ist ein Byte. Jede Speicherstelle des C 64 faßt genau ein solches Byte. Weil aber die größte in einem Byte darstellbare Zahl binär 11111111 ist — was dezimal 255 entspricht (rechnen Sie mal nach) —, meldet unser Computer einen ILLEGAL QUANTITY ERROR, wenn wir versuchen, eine größere Zahl in eine Speicherstelle zu POKEn.

Der C 64 ist darauf eingerichtet, ganze Zahlen als 2-Byte-Werte (also mit 16 Bit) zu verarbeiten. Die auf diese Weise größte darstellbare Zahl ist 11111111 11111111, was 65535 entspricht. Das ist übrigens genau die Anzahl der Speicherstellen unseres Computers. Allerdings erlaubt der C 64 bei Integerzahlen lediglich Eingaben zwischen 32767 und —32768. Die Ursache dafür ist die Art, wie negative Binärzahlen erzeugt und gespeichert werden.

e) Negative Binärzahlen

Bit 7 (also das ganz links stehende Bit eines Bytes: Die Zählung fängt rechts mit 0 an) einer 8-Bit-Zahl oder Bit 15 einer 16-Bit-Zahl wird als Vorzeichenkennung verwendet. Ist dieses Bit eine 0, dann liegt eine positive, ansonsten eine negative Zahl vor. Die Binärzahl 0111 1111 1111 1111 (32767) ist daher die größte vorzeichenbehaftete 16-Bit-Binärzahl. Wenn also Bit 7 (oder bei der 16-Bit-Zahl Bit 15) eine 1 enthält, handelt es sich um eine negative Zahl. Unser Computer hat aber eine besondere Art, solche negativen Zahlen zu verstehen: Er interpretiert sie als »Zweierkomplementzahlen«. Ein Beispiel, in dem wir die Zahl —29 in dieses Format bringen, soll erläutern, worum es sich dabei handelt:

Schritt 1:

Man berechnet zuerst die Binärzahl der positiven Zahl, also von 29:

0000 0000 0001 1101

Schritt 2:

Davon wird nun das sogenannte Einerkomplement gebildet, wobei jedes Bit in sein Gegenteil verkehrt wird:

1111 1111 1110 0010

Schritt 3:

Aus dem so erhaltenen Einerkomplement bildet man das »Zweierkomplement« durch Addieren einer 1:

```
1111   1111   1110   0010
                        1 +
--------------------------
1111   1111   1110   0011
```

Das Ergebnis ist —29 im Zweierkomplement-Format.

Hat man eine Zahl im Zweierkomplementformat vorliegen, dann kann man auf genau demselben Weg ihren Wert feststellen. Das wollen wir uns noch an einer 8-Bit-Zahl ansehen:

Schritt 1:

Es liegt eine Binärzahl vor, die so aussieht:

1111 1110

Schritt 2:

Das Einerkomplement wird gebildet:

0000 0001

Schritt 3:

Das Zweierkomplement wird durch Addieren von 1 erzeugt:

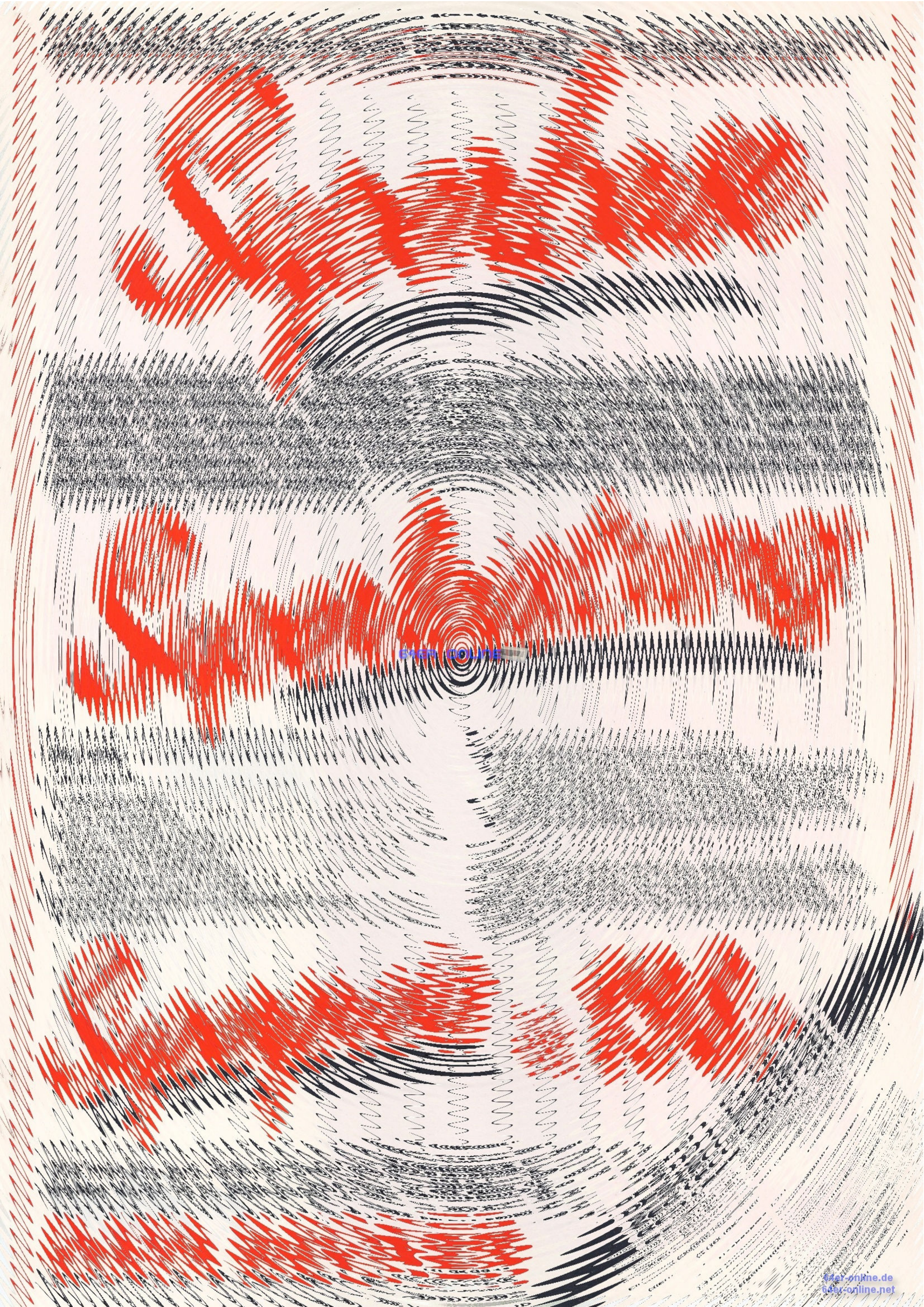
```
0000   0001
          1 +
-------------
0000   0010
```

Schritt 4:

Umrechnen der Binärzahl in die Dezimalzahl ergibt hier den Wert 2. Das Vorzeichen war negativ (Bit 7 ist ja 1 in der ursprünglichen Zahl). Es handelte sich also um die Zahl —2.

256	=0000	0001	0000	0000
Einerkomplement	= 1111	1110	1111	1111
+ 1				1
Zweierkomplement	= 1111	1111	0000	0000
= —256				
unsere Binärzahl			0101	0101
	1111	1111	0101	0101
NOT-Operation:	0000	0000	1010	1010
	↑ Vorzeichenbit = 0 (positiv)			

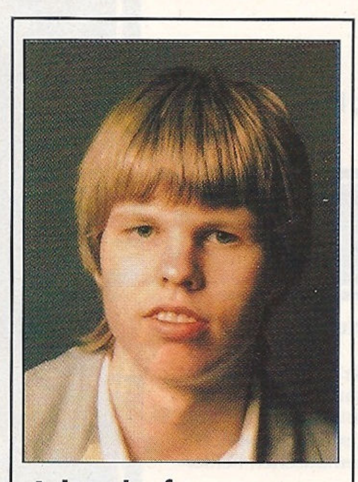
Bild 2. Ablauf der Operation NOT (A—256) bei A=0101 0101




64ER ONLINE

Terminalprogramm der Spitzenklasse

Eine harte Nuß haben wir unseren Lesern mit diesem Wettbewerb zu knacken gegeben. Trotzdem erreichten uns viele Einsendungen als Anwärter auf den ersten Preis.



Lebenslauf

Ich wurde am 15.9.1966 geboren. Heute gehe ich in die 12. Klasse des Ratsgymnasiums in Stadthagen. Bevor ich mich mit dem Computer beschäftigte, hatte ich die Hobbys Modelleisenbahn und Elektronik. Weiterhin bin ich noch Modellflieger.

Den C 64 habe ich seit einem Jahr. In dieser Zeit lernte ich zunächst Basic und Pascal, was auch bei uns in der Schule im Informatikunterricht verwendet wird. Seit einiger Zeit beschäftige ich mich mit Assembler.

Vor zirka einem halben Jahr kaufte ich mir einen Akustikkoppler. Da ich kein passendes Terminalprogramm fand, begann ich mir ein eigenes Programm zu schreiben. Dieses wurde dann immer mehr erweitert, bis es die heutige Form hatte.

(Peter Falk)

Peter Falk konnte mit seinem Programm »Super Term 64« diesen Wettbewerb für sich entscheiden. Besonders erwähnenswert ist neben der Leistung dieses Programmes der übersichtliche und gut durchschaubare Programmierstil.

Der Autor: »Ich habe bei dem Terminal-Programm einerseits darauf Wert gelegt, daß es möglichst komfortabel und benutzerfreundlich ist, andererseits war mir aber auch ein großer Terminal-Speicher sehr wichtig. Deswegen habe ich auf unnötige »Verzierungen« verzichtet. So kann man bei den Übertragungs-Parametern zum Beispiel nur Baudrate, Datenbits und Stop-Bits in begrenztem Umfang ändern. Ich habe jedoch die Erfahrung gemacht, daß man bei allen Mailboxen eine einwandfreie Verbindung hat, wenn man die Normalparameter (300 Baud, 7 Bit, 1 Stop-Bit) benutzt. Der C 64 empfängt nämlich auch dann richtig, wenn er zum Beispiel auf 7 Datenbit eingestellt ist und die Gegenstelle mit 8 Bit sendet. Nur wenn man auch Steuerzeichen übertragen will, muß man auf 8 Datenbit umstellen.

Der große Terminal-Speicher wurde unter anderem dadurch erreicht, daß der gesamte vom ROM überlagerte RAM genutzt wird, indem das Betriebssystem beim Zugriff auf den RAM mit einer Assembler-Routine abgeschaltet wird. Weitere Maschinen-Routinen werden für das Laden von Daten in den Terminal-Speicher und für das Listen des Directorys benutzt. Sie werden alle über DATA-Zeilen im Lader eingelesen, so daß sie nachher keinen Speicherplatz mehr verbrauchen.

Bei der Programm-Übertragung nach der »Teleterm-Norm« können natürlich die selben Fehler auftreten wie bei Teleterm, nämlich dann, wenn sich zwei Übertragungsfehler in einem Block gegenseitig aufheben. Trotzdem ist die Übertragungs-Routine etwas sicherer als bei Teleterm. Am Anfang der Übertragung kann man die Länge einer Übertragungseinheit einstellen (bei Teleterm: »Programmsatzlänge«). Bei einer guten Verbindung kann sie ruhig auf 256 Byte gestellt werden; die Übertragung ist dann etwas schneller.«

Achtung!

Die Qualitäten dieses Programmes zwangen uns zu einer etwas ungewöhnlichen Maßnahme.

Das hier abgedruckte Basic-Programm ist in dieser Form nicht lauffähig. Es muß erst mit dem Austro-Speed-Compiler compiliert werden. Was soll dann der Abdruck, beziehungsweise wer hat schon diesen Compiler, werden Sie wohl fragen. Nun, wir sind von diesem Terminalprogramm und seinen Leistungen überzeugt. Ein Abdruck der mit MSE einzugebenden compilierten Version würde mindestens zehn Seiten beanspruchen. Dies halten wir für nicht zumutbar. Um nun allen Lesern dieses Programm zugänglich zu machen, bieten wir in diesem Fall einen Extraservice. Alle, die an dem MSE-Listing interessiert sind, können dies gegen Einsendung eines Freiumschlages (DIN A6, frankiert mit 1,10 Mark) bei folgender Adresse anfordern:
Markt & Technik Verlag
Redaktion 64'er
Stichwort: Terminal
Hans-Pinsel-Straße 2
8013 Haar bei München

Eine weitere Möglichkeit, das Listing zu nutzen, bietet Ihnen natürlich auch der Leserservice. Auf der heftbegleitenden Diskette befindet sich natürlich auch die compilierte Version. (rg)

Ladeprogramm

Nachdem man den Lader geladen und mit RUN gestartet hat, werden die Funktionstasten-Belegungen von der Diskette geladen, soweit sie schon vorhanden sind. Ansonsten wird der F-Tasten-Speicher gelöscht. Danach erscheint ein Menü, in dem man zwischen dem Terminal-Programm und dem Editor wählen kann. Wenn man 1 oder 2 gedrückt hat, wird das jeweilige Programm geladen und automatisch gestartet.

In die ersten Programmzeilen des Laders kann man individuelle Parameter schreiben. So kann man in Zeile 90 Hintergrund- und Rahmenfarbe eingeben (Schriftfarbe muß schwarz bleiben), in Zeile 110 den Preis für eine Telefon-Gebühreneinheit ändern und in den Zeilen 30 und 40 kann man Parameter zur Druckeranpassung eingeben (Sekundäradresse und maximal fünf sonstige Parameter).

In den Zeilen 350 bis 580 werden die Assembler-Routinen für Speicherzugriff, LOAD und Directory eingelesen.

Die vorhandenen Druckerparameter sind für einen Epson-Drucker mit Görlitz-Interface zugeschnitten (Sekundäradresse: 0, Parameter: »chr$(27)"t"«, also 27, 32, 84).

Terminalprogramm

Das Terminal-Programm muß mit dem Lader geladen werden, damit es funktionstüchtig ist. Im folgenden werden die einzelnen Menüpunkte erklärt:

1. Kommunizieren

Nachdem man 1 gedrückt hat, muß man eingeben, ob die Gegenstelle mit oder ohne Echo arbeitet. Danach wird man gefragt, ob man eine Berechnung der Telefongebühren wünscht. Wenn ja, dann müssen die Gebührenzone und der Gebührentarif (vor oder nach 18 Uhr plus Wochenende) eingegeben werden. Nachdem man die Verbindung hergestellt hat (was ja durchaus etwas länger dauern kann!), startet man den Terminal-Modus mit einem Tastendruck (die ←-Taste führt ins Menü zurück). Im Terminal-Modus wird nun eine Help-Liste ausgegeben, in der alle Funktionen aufgeführt sind, die man



über »Commodore«-Taste ansprechen kann. **»Commodore S«** bewirkt, daß alle empfangenen Zeichen in den Terminal-Speicher geschrieben werden, der eine Größe von 46000 Byte hat. Auf ein automatisches Abspeichern habe ich wegen der Time-Out-Regelung der meisten Mailboxen verzichtet. Außerdem muß man schon ganz schön lange telefonieren, um den Speicher voll zu bekommen.

»Commodore E« beendet den Speichervorgang wieder.

»Commodore A« bewirkt, daß der Inhalt des Terminal-Speichers ausgesendet wird. So kann man zum Beispiel vorher abgefaßte Briefe übertragen. Abbruch mit '←'. Auch Basic-Programme, die in einer Sequenz vorliegen, können so übertragen werden. Diese Prozedur reagiert auf die Steuerzeichen

CTRL — S = Stop
CTRL — Q = Weiter
CTRL — X = Abbruch

»Commodore L« löscht den Terminal-Speicher.

»Commodore B« gibt an, wieviele Bytes im Speicher schon belegt und wieviele noch frei sind.

»Commodore G« zeigt die Zeit und die Gebühren an, die schon »vertelefoniert« sind.

»Commodore H« gibt die angezeigte Help-Liste nochmals aus.

»Commodore P« bietet eine Besonderheit: In diesem Modus ist es möglich Basic- und Maschinen-Programme zu übertragen und zwar nach der »TELETERM-Norm«. Man kann sich jetzt also Programme aus Mailboxen holen, die eine Teleterm-Box besitzen (zum Beispiel K.I.S., R.T.C., MHB etc.) oder man kann Programme unter Freunden und Bekannten austauschen, ohne Teleterm benutzen zu müssen. Es ist so eine weitgehend fehlerlose Übertragung von Programmen möglich.

Statt »Commodore £« und »1« bei Teleterm muß man hier »Commodore P« und »E« für empfangen drücken.

»Commodore ↑« unterbricht den Terminal-Modus, das heißt man kann jetzt alle Menüpunkte anwählen, ohne daß die eingestellten Parameter und die Gebühren verloren gehen. Man kann so also auch während man mit einer Mailbox kommuniziert, die anderen Menüpunkte ausführen.

»Commodore ←« beendet den Terminal-Modus. Hier werden jetzt die Zeit und die Gebühren angezeigt und abgespeichert, und dann der Gesamtbetrag angezeigt.

2. Daten speichern

Hier wird der Inhalt des Terminal-Speichers auf Diskette gespeichert. Man muß dafür den Namen angeben, unter dem abgespeichert werden soll.

RETURN bewirkt den Rücksprung ins Menüs.

3. Entspricht Punkt 2, nur daß hier geladen wird.

4. Daten ausgeben

Unter diesem Menüpunkt wird der Inhalt des Terminal-Speichers auf dem Bildschirm oder dem Drucker ausgegeben.

F1 = Ausgabe auf dem Bildschirm
F3 = Ausgabe auf dem Drucker
F5 = Bildschirm-Hardcopy
← = Menü

SPACE hält die Ausgabe an. Sie kann dann mit einer beliebigen Taste fortgesetzt werden.

5. Funktionstasten

Hier kann man sich eine Übersicht der F-Tasten-Belegungen ausgeben lassen und sie gegebenenfalls ändern und abspeichern.

6. Parameter ändern

Die Baudrate und die Anzahl der Daten- und Stop-Bits können hier verändert werden.

7. Gebühren ansehen

Wenn man mit der Gebühren-Berechnung arbeitet, kann man hier den abgespeicherten Gesamtbetrag abrufen und die Gebühren am Ablesetag löschen.

8. Diskbefehle

Unter diesem Punkt kann man Befehle wie zum Beispiel Validate, Scratch, Rename etc. der Floppy übermitteln. Mit dem Dollarzeichen kann man das Inhaltsverzeichnis auflisten.

Editor-Programm

Mit dem Editor kann man Anzeigen oder Briefe, die man in Mailboxen schreiben will, erstellen. Der Text wird nach dem Erstellen sofort in den Terminal-Speicher geschrieben, so daß er nicht erst auf Diskette gespeichert zu werden braucht, bevor man das Terminal-Programm lädt. Der Speicherplatz für Variablen ist deswegen nur so groß, daß der Editor maximal 100 Zeilen aufnehmen kann.

Wenn man den Menüpunkt 1 anwählt, wird man zuerst gefragt, ob der Terminal-Speicher gelöscht werden soll. Antwortet man mit »ja«, so kann man einen neuen Text schreiben. Ansonsten wird der Inhalt des Terminal-Speichers in den Editor übertragen. Hier kann man nun seinen Text eingeben und mit den Cursor-Tasten frei editieren. Man muß allerdings in jeder Zeile, die übernommen werden soll, RETURN drücken. Wenn der Cursor am Anfang einer Zeile steht, kann man vor dieser Zeile mit CTRL-I (Insert) eine Zeile einfügen und mit CTRL-D (Delete) kann man die Zeile löschen. Wenn die maximale Anzahl von 100 Zeilen erreicht ist, wechselt die Rahmenfarbe in Rot und es wird keine weitere Zeile mehr übernommen.

Die ←-Taste führt ins Menü zurück, nachdem der Text in den Terminal-Speicher geschrieben wurde.

Die anderen Menüpunkte entsprechen denen des Terminal-Programms.

(Peter Falk/rg)

```
10 REM WRITTEN BY PETER FALK                <255>
20 :                                        <078>
30 POKE 840,0:REM SEKUNDAER-ADRESSE         <076>
40 POKE 841,27:POKE 842,32:POKE 843,84:POK
   E 844,0:POKE 845,0:REM DRUCKER-PARAMETE
   R                                        <059>
50 :                                        <108>
60 POKE 55,79:POKE 56,60:REM SPEICHER-      <240>
70 POKE 643,79:POKE 644,60:REM GRENZE       <171>
80 CLR                :REM AUF 15439        <151>
90 POKE 838,6:POKE 839,15:REM FARBEN        <116>
100 POKE 53280,PEEK(838):POKE 53281,PEEK(8
    39)                                     <150>
110 G%=100*0.23:POKE 828,G%:REM GEBUEHREN-
    EINHEIT                                 <040>
120 PRINT"{BLACK,CLR,DOWN}";CHR$(14);CHR$(
    8)                                      <204>
130 GOSUB 350:GOSUB 600:POKE 650,128        <055>
140 POKE 832,80:REM PUFFERANFANG            <144>
150 POKE 833,60:REM BEI 15440               <244>
160 PRINT"{CLR,2DOWN,5SPACE}***************
    *****************"                      <072>
170 PRINT"{5SPACE}*{29SPACE}*"              <219>
180 PRINT"{5SPACE}*{3SPACE}S U P E R - T E
    R M{2SPACE}64{3SPACE}*"                 <195>
190 PRINT"{5SPACE}*{3SPACE}YYYYYYYYYYYYYYY
    YYYYYYYY{3SPACE}*"                      <096>
200 PRINT"{5SPACE}*{29SPACE}*"              <249>
210 PRINT"{5SPACE}*{4SPACE}WRITTEN BY PETE
    R FALK{4SPACE}*"                        <233>
220 PRINT"{5SPACE}*{29SPACE}*"              <013>
230 PRINT"{5SPACE}*{6SPACE}IN THE YEAR 198
    5{7SPACE}*"                             <023>
240 PRINT"{5SPACE}*{29SPACE}*"              <033>
250 PRINT"{5SPACE}*************************
    *******"                                <237>
260 PRINT"{2DOWN,8SPACE}(1) - TERMINALPROG
    RAMM{DOWN}"                             <005>
270 PRINT"{8SPACE}(2) - EDITOR{2DOWN}"      <005>
280 PRINT"{8SPACE}(0) - PROGRAMM BEENDEN"   <122>
290 GET Q$:IF Q$<>"0"AND Q$<>"1"AND Q$<>"2
    "THEN 290                               <136>
300 IF Q$="0"THEN END                       <122>
310 PRINT"{2DOWN,14SPACE}LOADING..."        <062>
320 IF Q$="1"THEN LOAD"TERMINAL",8          <166>
330 IF Q$="2"THEN LOAD"EDITOR",8            <028>
340 :                                       <143>
350 REM * PEEK+POKE FUER UEBERL. RAM *      <192>
360 DATA 165,20,72,165,21,72,32,247,183,12
    0,165,1,41,252,133,1,160,0,177,20       <139>
370 DATA 168,165,1,9,3,133,1,88,104,133,21
    ,104,133,20,76,162,179,32,253,174       <169>
380 DATA 32,138,173,32,247,183,32,253,174,
    32,158,183,120,169,52,133,1,138,160     <033>
390 DATA 0,145,20,169,55,133,1,88,96        <071>
400 FOR I=698 TO 765:READ A:POKE I,A:NEXT   <063>
410 POKE 785,186:POKE 786,2                 <117>
420 :                                       <223>
430 REM * LOAD *                            <178>
440 DATA 160,0,169,8,32,180,255,169,98,32,
    150,255,173,64,3,133,174,173,65         <166>
450 DATA 3,133,175,32,19,238,145,174,230,1
    74,208,2,230,175,169,208,197,175        <226>
460 DATA 208,8,169,0,133,174,169,224,133,1
    75,165,175,240,4,36,144,80,223          <129>
470 DATA 165,174,141,64,3,165,175,141,65,3
    ,32,171,255,96,0                        <211>
480 FOR I=848 TO 917:READ A:POKE I,A:NEXT   <139>
490 :                                       <037>
500 REM * DIRECTORY *                       <141>
510 DATA 169,36,133,251,169,251,133,187,16
    9,0,133,188,169,1,133,183,169,8         <249>
520 DATA 133,186,169,96,133,185,32,213,243
    ,165,186,32,180,255,165,185,32,150      <145>
530 DATA 255,169,0,133,144,160,3,132,251,3
    2,165,255,133,252,164,144,208,47        <029>
540 DATA 32,165,255,164,144,208,40,164,251
    ,136,208,233,166,252,32,205,189         <003>
550 DATA 169,32,32,210,255,32,165,255,166,
    144,208,18,170,240,6,32,210,255         <254>
560 DATA 76,226,3,169,13,32,210,255,160,2,
    208,198,32,66,246,96                    <252>
570 FOR I=918 TO 1022:READ A:POKE I,A:NEXT  <008>
580 RETURN                                  <212>
590 :                                       <138>
600 REM * F-TASTEN LADEN *                  <227>
610 OPEN 1,8,2,"F-TASTEN,S,R"               <165>
620 OPEN 2,8,15                             <053>
630 INPUT#2,F                               <160>
640 IF F=0 THEN PRINT"{3DOWN,2SPACE}DIE FU
    NKTIONSTASTEN WERDEN GELADEN."          <049>
650 PRINT"{3DOWN,12SPACE}BITTE WARTEN..."   <239>
660 IF F=0 THEN 700                         <135>
670 FOR S=54000 TO 54559                    <207>
680 SYS 735,S,13:NEXT                       <178>
690 GOTO 720                                <214>
700 FOR S=54000 TO 54559                    <237>
710 GET#1,A$:SYS 735,S,ASC(A$):NEXT         <168>
720 CLOSE 1:CLOSE 2                         <175>
730 RETURN                                  <106>
```

Listing »Lader«. Bitte beachten Sie die Eingabehinweise auf Seite 77.

```
10 REM WRITTEN BY PETER FALK                <255>
20 REM 828=PREIS EINER EINHEIT              <126>
30 POKE 829,32:REM 7 DATENBITS              <135>
40 POKE 830,6:REM 300 BAUD                  <052>
50 POKE 831,0:REM 1 STOPBIT                 <224>
60 REM 832,80:REM (LO) PUFFERANFANG         <121>
70 REM 833,60:REM (HI) BEI 15440            <211>
80 REM 834=GEGENSTELLE DUPLEX               <188>
90 REM 835=GEBUEHREN JA/NEIN                <063>
100 REM 836=GEBUEHREN-ZONE                  <207>
110 REM 837=GEBUEHREN-TARIF                 <020>
120 REM 838=RAHMEN-FARBE                    <047>
130 REM 839=HINTERGRUND-FARBE               <201>
140 REM 840-845=DRUCKER-PARAMETER           <160>
150 :                                       <208>
160 REM ** MENUE **                         <081>
170 PRINT:PRINT"{CLR,BLACK,2DOWN,7SPACE}PP
    PPPPPPPPPPPPPPPPPPPPPP"                 <007>
180 PRINT"{7SPACE,RVSON,2SPACE}** TERMINAL
    -MENUE **{2SPACE}"                      <206>
190 PRINT"{7SPACE}YYYYYYYYYYYYYYYYYYYYYYYY
    {2DOWN}"                                <229>
200 PRINT"{7SPACE}(1) - KOMMUNIZIEREN";A$;
    "{DOWN}"                                <241>
210 PRINT"{7SPACE}(2) - DATEN SPEICHERN{DO
    WN}"                                    <253>
220 PRINT"{7SPACE}(3) - DATEN LADEN{DOWN}" <203>
230 PRINT"{7SPACE}(4) - DATEN AUSGEBEN{DOW
    N}"                                     <188>
240 PRINT"{7SPACE}(5) - FUNKTIONSTASTEN{DO
    WN}"                                    <161>
250 PRINT"{7SPACE}(6) - PARAMETER{DOWN}"   <189>
260 PRINT"{7SPACE}(7) - GEBUEHREN{DOWN}"   <189>
270 PRINT"{7SPACE}(8) - DISKBEFEHLE{DOWN}" <073>
280 PRINT"{7SPACE}(9) - EDITOR LADEN"      <088>
290 GET Q$:IF Q$="" THEN 290                <171>
300 IF Q$="1" AND A$="(FORTSETZUNG)" THEN
    A$="":GOTO 480                          <251>
310 ON VAL(Q$) GOTO 360,2390,2610,2770,411
    0,3070,3850,4810,340                    <000>
320 IF Q$="0" THEN END                      <142>
330 GOTO 290                                <111>
340 LOAD"EDITOR",8                          <087>
350 :                                       <153>
360 REM *** KOMMUNIZIEREN ***               <225>
370 INPUT"{CLR,2DOWN,2SPACE}GEGENSTELLE{SP
    ACE,RVSON}H{RVOFF}ALB- /{SPACE,RVSON}V
    {RVOFF}OLLDUPLEX{3SPACE}V{3LEFT}";Q$    <071>
380 IF Q$="V" THEN POKE 834,1               <003>
390 IF Q$="H" THEN POKE 834,0               <254>
400 PRINT"{2DOWN,2SPACE}GEBUEHREN-BERECHNU
    NG (J/N)?{DOWN}"                        <155>
410 GET Q$:IF Q$<>"J" AND Q$<>"N" THEN 410 <051>
420 POKE 835,0                              <056>
430 IF Q$="J" THEN POKE 835,1:GOSUB 3390   <192>
440 PRINT"{2DOWN,11SPACE}ZUM STARTEN TASTE
     !"                                     <247>
450 GET Q$:IF Q$="" THEN 450                <073>
460 IF Q$="←" THEN 160                      <096>
470 TI$="000000"                            <174>
480 BAUD=PEEK(830):DB=PEEK(829):SB=PEEK(83
    1)                                      <181>
490 OPEN 2,2,0,CHR$(BAUD+DB+SB)+CHR$(0):GE
    T#2,Q$                                  <008>
500 PRINT"{CLR}";                           <160>
510 GOSUB 4990                              <098>
520 GOSUB 4030                              <094>
530 DP=-PEEK(834)                           <183>
540 GB=-PEEK(835)                           <183>
550 GET#2,Q$:PRINT                          <146>
560 :                                       <108>
570 GOSUB 4700                              <148>
580 GET B$:IF B$="" THEN 650                <176>
590 GOSUB 4760                              <174>
600 IF ASC(B$)>132 AND ASC(B$)<141 THEN GO
    SUB 4480:GOTO 570                       <129>
610 IF PEEK(653)=2 THEN 2250                <244>
620 GOSUB 2100                              <190>
630 :                                       <178>
640 GOSUB 4700                              <218>
650 IF PEEK(667)=PEEK(668) THEN 580         <122>
660 GOSUB 4760                              <244>
670 GET#2,C$:IF C$="" THEN C$=CHR$(0)       <083>
680 GOSUB 2170                              <001>
690 IF NOT SP THEN 570                      <205>
700 POKE SG,D:SG=SG+1:IF SG=53248 THEN SG=
    57344                                   <107>
710 IF SG=65536 THEN POKE 53280,PEEK(838):
    SP=0:SG=65535                           <023>
720 GOTO 570                                <247>
730 :                                       <022>
740 REM * PROGRAMM-UEBERTRAGUNG *           <230>
750 GOSUB 3970:CLOSE 2:CLR                  <050>
760 BAUD=PEEK(830)                          <118>
770 OPEN 2,2,0,CHR$(BAUD)+CHR$(0):GET#2,Q$ <179>
780 INPUT"{CLR,3DOWN}PROGRAMM-NAME:";DN$:I
    F DN$="" THEN PRINT:CLOSE 2:GOTO 480    <152>
790 DIM B$(258)                             <245>
800 OPEN 1,8,15,"M-W"+CHR$(7)+CHR$(28)+CHR
    $(1)+CHR$(15):CLOSE 1                   <085>
810 PRINT"{2DOWN,RVSON}S{RVOFF}ENDEN /{SPA
    CE,RVSON}E{RVOFF}MPFANGEN ?"            <048>
820 GET Q$:IF Q$<>"S" AND Q$<>"E" THEN 820 <212>
830 IF Q$="E" THEN 1440                     <236>
840 :                                       <133>
850 REM * SENDEN *                          <245>
860 PRINT"{2DOWN}LAENGE EINER UEBERTRAGUNG
    S-EINHEIT"                              <014>
870 INPUT"IN BYTES (MAX.256){3SPACE}80{4LE
    FT}";E                                  <046>
880 IF E<1 OR E>256 THEN 860                <175>
890 OPEN 1,8,2,DN$+",P,R"                   <217>
900 GOSUB 4930                              <228>
910 IF VAL(B$)<>0 THEN 1360                 <157>
920 POKE 668,PEEK(667):POKE 198,0           <135>
930 PRINT#2,CHR$(2);                        <032>
940 GET#2,E$:GET Q$                         <138>
950 FOR W=1 TO 100:NEXT                     <101>
960 IF E$="" AND Q$<>"←" THEN 930           <027>
970 IF Q$="←" THEN 1360                     <147>
980 PRINT"{DOWN}BEGINN DER UEBERTRAGUNG.{D
    OWN}"                                   <029>
990 FOR W=1 TO 250:NEXT                     <147>
1000 POKE 688,PEEK(667):BY=0                <193>
1010 :                                      <047>
1020 FOR Z=1 TO E                           <166>
1030 GET#1,B$(Z):IF B$(Z)="" THEN B$(Z)=CH
     R$(0)                                  <186>
1040 IF ST=0 THEN NEXT Z                    <171>
1050 F=ST:IF Z>E THEN Z=E                   <202>
1060 FOR W=1 TO 1000:NEXT                   <004>
1070 SU=0:GET#2,E$                          <129>
1080 FOR X=1 TO Z                           <246>
1090 PRINT#2,B$(X);                         <134>
1100 SU=SU+ASC(B$(X))                       <034>
1110 NEXT X                                 <052>
1120 IF PEEK(669)<>PEEK(670) THEN 1120      <038>
1130 POKE 668,PEEK(667)                     <139>
1140 GET Q$:IF PEEK(667)=PEEK(668) AND Q$<
     >"←" THEN 1140                         <010>
1150 IF Q$="←" THEN 1360                    <072>
1160 GET#2,LO$:IF LO$="" THEN LO$=CHR$(0)   <071>
1170 GET Q$:IF PEEK(667)=PEEK(668) AND Q$<
     >"←" THEN 1170                         <043>
1180 IF Q$="←" THEN 1360                    <102>
1190 GET#2,HI$:IF HI$="" THEN HI$=CHR$(0)   <071>
1200 SE=ASC(LO$)+256*ASC(HI$)               <147>
1210 IF SE<>SU THEN 1300                    <088>
1220 BY=BY+Z                                <180>
1230 PRINT" OK.{7SPACE}";INT(BY/254)+1;"{L
     EFT}. BLOCK"                           <171>
1240 IF F<>0 THEN 1360                      <178>
1250 PRINT#2,CHR$(2);                       <097>
1260 GET Q$:IF PEEK(667)=PEEK(668) AND Q$<
     >"←" THEN 1260                         <133>
1270 IF Q$="←" THEN 1360                    <192>
1280 GOTO 1020                              <081>
1290 :                                      <073>
1300 REM ERROR                              <050>
1310 PRINT#2,CHR$(1);:PRINT" ERROR !"       <223>
1320 GET Q$:IF PEEK(667)=PEEK(668) AND Q$<
     >"←" THEN 1320                         <191>
1330 FOR W=1 TO 1000:NEXT                   <019>
1340 IF Q$<>"←" THEN 1070                   <183>
1350 :                                      <133>
1360 IF Q$="←" THEN PRINT"{DOWN}ABBRUCH !{
     DOWN}"                                 <232>
1370 PRINT#2,CHR$(3);                       <219>
1380 PRINT"{DOWN}ENDE DER UEBERTRAGUNG."    <007>
1390 CLOSE 1:CLOSE 15:CLOSE 2               <144>
1400 PRINT"{2DOWN,11SPACE}TASTE DRUECKEN !
     "                                      <239>
1410 WAIT 198,255:POKE 198,0                <082>
1420 GOTO 480                               <182>
1430 :                                      <213>
1440 REM * EMPFANGEN *                      <025>
1450 OPEN 1,8,2,DN$+",P,W"                  <016>
1460 GOSUB 4930                             <022>
1470 IF VAL(B$)<>0 THEN 1800                <206>
1480 POKE 668,PEEK(667):POKE 198,0          <185>
1490 PRINT#2,CHR$(2);                       <082>
1500 GET#2,E$:GET Q$                        <188>
1510 FOR W=1 TO 100:NEXT                    <151>
1520 IF E$="" AND Q$<>"←" THEN 1490         <127>
1530 IF Q$="←" THEN 1800                    <196>
1540 PRINT"{DOWN}BEGINN DER UEBERTRAGUNG.{
     DOWN}"                                 <080>
1550 FOR W=1 TO 100:NEXT                    <192>
1560 GET#2,E$:POKE 668,PEEK(667):BY=0       <183>
1570 SU=0:Z=1                               <041>
1580 GET Q$:IF PEEK(667)=PEEK(668) AND Q$<
     >"←" THEN 1580                         <204>
1590 IF Q$="←" THEN 1800                    <001>
1600 AT=TI                                  <042>
1610 IF TI-AT>30 THEN 1660                  <064>
1620 IF PEEK(667)=PEEK(668) THEN 1610       <115>
1630 GET#2,B$(Z):IF B$(Z)="" THEN B$(Z)=CH
     R$(0)                                  <021>
1640 SU=SU+ASC(B$(Z)):Z=Z+1                 <189>
1650 IF Z<258 THEN 1600                     <029>
1660 HI=INT(SU/256):LO=SU-256*HI            <113>
1670 PRINT#2,CHR$(LO);CHR$(HI);             <084>
1680 GET Q$:IF PEEK(667)=PEEK(668) AND Q$<
     >"←" THEN 1680                         <049>
1690 IF Q$="←" THEN 1800                    <101>
1700 GET#2,E$                               <053>
1710 IF E$<>CHR$(2)AND E$<>CHR$(3)THEN 187
     0                                      <148>
1720 PRINT#2,CHR$(4);                       <059>
1730 FOR X=1 TO Z                           <130>
1740 PRINT#1,B$(X);                         <017>
1750 NEXT X                                 <182>
1760 GET#2,Q$                               <125>
1770 BY=BY+Z                                <220>
1780 PRINT" OK.{7SPACE}";INT(BY/254)+1;"{L
     EFT}. BLOCK"                           <211>
1790 IF E$=CHR$(2) THEN 1570                <104>
1800 IF Q$="←" THEN PRINT"{DOWN}ABBRUCH !{
     DOWN}"                                 <162>
1810 PRINT"{DOWN}ENDE DER UEBERTRAGUNG."    <183>
1820 CLOSE 1:CLOSE 15:CLOSE 2               <064>
1830 PRINT"{2DOWN,11SPACE}TASTE DRUECKEN !
     "                                      <159>
1840 WAIT 198,255:POKE 198,0                <002>
1850 GOTO 480                               <102>
1860 :                                      <133>
1870 REM ERROR                              <110>
1880 PRINT" ERROR !"                        <103>
1890 PRINT#2,CHR$(3);                       <229>
1900 POKE 668,PEEK(667)                     <144>
1910 GOTO 1570                              <211>
1920 :                                      <193>
1930 REM * DATEN SENDEN *                   <157>
1940 FOR SZ=15440 TO SG                     <183>
1950 GET Q$:IF Q$="←" THEN PRINT:RETURN     <082>
1960 IF Q$=" " THEN GET E$:IF E$<>" " THEN
      1960                                  <163>
1970 FOR W=1 TO 24:NEXT                     <058>
1980 GET#2,C$:IF C$=CHR$(24) THEN PRINT:RE
     TURN                                   <176>
1990 IF C$<>CHR$(19) THEN 2010              <015>
2000 GET#2,Q$:IF Q$<>CHR$(17) AND PEEK(203
     )=64 THEN 2000                         <104>
2010 SYS 62595:REM CIA'S SETZEN             <115>
2020 B$=CHR$(USR(SZ)):GET#2,C$              <147>
2030 IF DP THEN POKE 646,0:PRINT B$;        <194>
2040 IF SZ=53247 THEN SZ=57343              <250>
2050 GOSUB 2100                             <090>
2060 IF D=13 THEN FOR W=1 TO 100:NEXT       <076>
2070 NEXT SZ                                <077>
2080 RETURN                                 <150>
2090 :                                      <108>
2100 REM * UMFORMEN/SENDEN *                <116>
2110 E=0:D=ASC(B$):IF D>64 AND D<91 THEN E
     =32                                    <081>
2120 IF D=20 THEN D=8                       <008>
2130 D=D+E                                  <131>
2140 PRINT#2,CHR$(D);:IF NOT DP THEN POKE
     646,0:PRINT B$;                        <160>
2150 RETURN                                 <252>
2160 :                                      <178>
2170 REM * UMFORMEN/AUSGEBEN *              <071>
2180 E=0:D=ASC(C$):IF D>64 AND D<91 THEN E
     =32                                    <152>
2190 IF D>96 AND D<123 THEN E=-32           <111>
2200 IF D=8 THEN D=20                       <088>
2210 D=D+E                                  <211>
2220 POKE 646,0:POKE 199,0:PRINT CHR$(D);   <129>
2230 RETURN                                 <076>
2240 :                                      <002>
2250 REM * TASTATUR-AUSWERTUNG *            <111>
2260 E=ASC(B$)                              <080>
2270 IF E=174 AND SG<65536 THEN SP=-1:POKE
      53280,2:GOTO 570                      <113>
2280 IF E=182 THEN SG=15440:GOSUB 3970:GOT
     O 570                                  <247>
2290 IF E=177 THEN SP=0:POKE 53280,PEEK(83
     8):GOTO 570                            <091>
2300 IF E=176 AND SG>15440 THEN GOSUB 1930
     :GOTO 570                              <124>
2310 IF E=165 THEN GOSUB 3520:GOTO 570      <138>
2320 IF E=191 THEN GOSUB 4050:GOTO 570      <146>
2330 IF E=175 THEN 740                      <132>
2340 IF E=95 THEN POKE 53280,PEEK(838):GOS
     UB 3970:GOSUB 3520:CLOSE 2:WAIT 198,1
     :GOTO 160                              <155>
2350 IF E=222 THEN POKE 53280,PEEK(838):GO
     SUB 3970:CLOSE 2:A$="(FORTSETZUNG)":G
     OTO 160                                <066>
2360 IF E=180 THEN GOSUB 4990:GOTO 570      <197>
2370 GOTO 570                               <112>
2380 :                                      <143>
2390 REM *** DATEN SPEICHERN ***            <247>
2400 PRINT"{CLR}"                           <217>
2410 GOSUB 4030                             <199>
2420 IF SG=15440 THEN 160                   <144>
2430 DN$=""                                 <051>
2440 INPUT"{2DOWN}DATEINAME ";DN$           <117>
2450 IF DN$="" THEN 160                     <016>
2460 OPEN 1,8,2,DN$+",S,W"                  <009>
2470 GOSUB 4930                             <012>
2480 PRINT#1,CHR$(USR(15440));              <007>
2490 IF ST<>0 THEN 2570                     <254>
2500 CMD 1:A=15441                          <199>
2510 E=SG:IF E>53247 THEN E=53247           <203>
2520 FOR SZ=A TO E                          <235>
2530 PRINT CHR$(USR(SZ));                   <140>
2540 NEXT                                   <119>
2550 IF E<SG THEN A=57344:E=SG:GOTO 2520    <020>
2560 PRINT#1,"";                            <126>
2570 CLOSE 1:CLOSE 15                       <037>
2580 IF VAL(B$)<>0 THEN WAIT 198,255        <092>
2590 GOTO 160                               <072>
2600 :                                      <108>
2610 REM *** DATEN LADEN ***                <151>
2620 DN$=""                                 <242>
2630 INPUT"{CLR,3DOWN}DATEINAME ";DN$       <216>
2640 IF DN$="" THEN 160                     <207>
2650 SG=15440                               <174>
2660 OPEN 1,8,2,DN$+",S,R"                  <205>
2670 GOSUB 4930                             <213>
2680 IF VAL(B$)<>0 THEN 2720                <143>
2690 POKE 144,0:POKE 832,80:POKE 833,60     <027>
2700 SYS 848:REM LOAD-ROUTINE               <020>
2710 SG=PEEK(832)+PEEK(833)*256-1           <028>
2720 CLOSE 1:CLOSE 15                       <187>
2730 IF SG=-1 THEN PRINT"{2DOWN}ACHTUNG, P
     UFFER VOLL !!!{2DOWN}":SG=65535        <091>
2740 GOSUB 3970:POKE 198,0:WAIT 198,255     <035>
2750 GOTO 160                               <232>
2760 :                                      <012>
2770 REM *** DATEN AUSGEBEN ***             <029>
2780 PRINT"{CLR}"                           <086>
2790 GOSUB 4030                             <068>
2800 IF SG=15440 THEN 160                   <013>
2810 PRINT"{DOWN}'F1' = BILDSCHIRM"         <213>
2820 PRINT"'F3' = DRUCKER"                  <000>
2830 PRINT"'F5' = HARDCOPY{DOWN}"           <103>
2840 WAIT 198,255                           <031>
2850 D=0                                    <083>
2860 FOR SZ=15440 TO SG                     <083>
2870 IF PEEK(198)<>0 THEN 2980              <143>
2880 C$=CHR$(USR(SZ))                       <049>
2890 PRINT C$;                              <144>
2900 IF D THEN PRINT#1,C$;                  <108>
2910 IF SZ=53247 THEN SZ=57343              <100>
2920 NEXT SZ                                <162>
2930 CLOSE 1:POKE 53280,PEEK(838)           <003>
2940 GET B$:IF B$="" THEN 2940              <037>
2950 IF B$="{F5}" THEN GOSUB 5130           <252>
2960 GOTO 160                               <187>
2970 :                                      <223>
2980 GET B$                                 <182>
2990 IF B$=" " THEN WAIT 198,255:GET B$     <132>
3000 IF B$="←" THEN CLOSE 1:POKE 53280,PEE
     K(838):GOTO 160                        <144>
3010 IF B$="{F3}" AND D=0 THEN E=PEEK(840)
     :POKE 53280,2                          <141>
3020 IF B$="{F3}"AND D=0 THEN OPEN 1,4,E:F
     OR E=841 TO 845:PRINT#1,CHR$(PEEK(E))
     ;:NEXT:D=1                             <236>
3030 IF B$="{F1}" THEN D=0:CLOSE 1:POKE 53
     280,PEEK(838)                          <218>
3040 IF B$="{F5}" THEN GOSUB 5130           <086>
3050 GOTO 2880                              <080>
3060 :                                      <057>
3070 REM ** PARAMETER **                    <225>
3080 PRINT"{CLR,2DOWN}"                     <166>
3090 :                                      <088>
```

Listing »Terminal«. Bitte beachten Sie bei der Eingabe den Checksummer 64.

```
3100 REM BAUDRATE                              <255>
3110 IF PEEK(830)=3 THEN BAUD=110              <087>
3120 IF PEEK(830)=5 THEN BAUD=150              <103>
3130 IF PEEK(830)=6 THEN BAUD=300              <111>
3140 IF PEEK(830)=7 THEN BAUD=600              <125>
3150 PRINT SPC(22) BAUD;"{UP}"                 <082>
3160 INPUT"110,150,300,600 BAUD ";BAUD         <245>
3170 IF BAUD=110 THEN POKE 830,3               <145>
3180 IF BAUD=150 THEN POKE 830,5               <161>
3190 IF BAUD=300 THEN POKE 830,6               <169>
3200 IF BAUD=600 THEN POKE 830,7               <183>
3210 :                                         <208>
3220 REM DATENBITS                             <205>
3230 IF PEEK(829)=32 THEN DB=7                 <024>
3240 IF PEEK(829)=0 THEN DB=8                  <238>
3250 PRINT"{3DOWN}";SPC(20) DB;"{UP}"          <208>
3260 INPUT"7 ODER 8 DATENBITS ";DB             <009>
3270 IF DB=7 THEN POKE 829,32                  <062>
3280 IF DB=8 THEN POKE 829,0                   <020>
3290 :                                         <032>
3300 REM STOPBITS                              <247>
3310 IF PEEK(831)=0 THEN SB=1                  <053>
3320 IF PEEK(831)=128 THEN SB=2                <171>
3330 PRINT"{3DOWN}";SPC(19) SB;"{UP}"          <056>
3340 INPUT"1 ODER 2 STOPBITS ";SB              <055>
3350 IF SB=1 THEN POKE 831,0                   <092>
3360 IF SB=2 THEN POKE 831,128                 <178>
3370 GOTO 160                                  <087>
3380 :                                         <123>
3390 REM ** GEBUEHREN BERECHNEN **             <161>
3400 PRINT"{DOWN,2SPACE}0. ZONE: NAHGESPRA
     ECH"                                      <128>
3410 PRINT"{DOWN,2SPACE}1. ZONE: UNTER 50
     KM"                                       <045>
3420 PRINT"{DOWN,2SPACE}2. ZONE: 50 - 100
     KM"                                       <104>
3430 PRINT"{DOWN,2SPACE}3. ZONE: UEBER 100
      KM"                                      <084>
3440 ZO=4:INPUT"{DOWN,2SPACE}ZONE 0, 1, 2
     ODER 3 ";ZO                               <008>
3450 IF ZO<>0 AND ZO<>1 AND ZO<>2 AND ZO<>
     3 THEN 3440                               <139>
3460 INPUT"{DOWN,2SPACE,RVSON}N{RVOFF}ORMA
     L- /{SPACE,RVSON}B{RVOFF}ILLIG-TARIF
     ";Q$                                      <094>
3470 IF Q$<>"N" AND Q$<>"B" THEN 3460          <019>
3480 POKE 836,ZO                               <178>
3490 POKE 837,ASC(Q$)                          <160>
3500 RETURN                                    <071>
3510 :                                         <253>
3520 IF NOT GB THEN RETURN                     <190>
3530 ZO=PEEK(836)                              <230>
3540 Q$=CHR$(PEEK(837))                        <213>
3550 ZEIT=VAL(LEFT$(TI$,2))*3600+VAL(MID$(
     TI$,3,2))*60+VAL(RIGHT$(TI$,2))           <000>
3560 IF Q$="B" THEN 3620                       <159>
3570 IF ZO=0 THEN BETRAG=INT(ZEIT/480+1)*P
     EEK(828)/100                              <040>
3580 IF ZO=1 THEN BETRAG=INT(ZEIT/45+1)*PE
     EK(828)/100                               <000>
3590 IF ZO=2 THEN BETRAG=INT(ZEIT/20+1)*PE
     EK(828)/100                               <005>
3600 IF ZO=3 THEN BETRAG=INT(ZEIT/12+1)*PE
     EK(828)/100                               <017>
3610 GOTO 3650                                 <127>
3620 IF ZO=0 THEN BETRAG=INT(ZEIT/720+1)*P
     EEK(828)/100                              <088>
3630 IF ZO=1 THEN BETRAG=INT(ZEIT/67.5+1)*
     PEEK(828)/100                             <154>
3640 IF ZO=2 OR ZO=3 THEN BETRAG=INT(ZEIT/
     38.6+1)*PEEK(828)/100                     <226>
3650 PRINT                                     <233>
3660 PRINT"ZEIT{2SPACE}=";TI$                  <044>
3670 PRINT"BETRAG=";BETRAG;"DM"                <179>
3680 PRINT                                     <007>
3690 IF B$<>"←" THEN 3830                      <229>
3700 PRINT"GEBUEHREN SPEICHERN (J/N)?"         <108>
3710 GET Q$                                    <162>
3720 IF Q$<>"J" AND Q$<>"N" THEN 3710          <020>
3730 IF Q$="N" THEN 3830                       <089>
3740 OPEN 1,8,2,"GEBUEHREN,S,R"                <062>
3750 INPUT#1,GESB                              <182>
3760 CLOSE 1                                   <143>
3770 GESB=GESB+BETRAG                          <027>
3780 OPEN 1,8,2,"@:GEBUEHREN,S,W"              <229>
3790 PRINT#1,GESB                              <242>
3800 CLOSE 1                                   <183>
3810 PRINT"{DOWN}GESAMTBETRAG=";GESB;"DM"      <125>
3820 WAIT 198,255                              <246>
3830 POKE 198,1:RETURN                         <098>
3840 :                                         <073>
3850 REM ** GEBUEHREN ANSEHEN **               <231>
3860 PRINT"{CLR,2DOWN}"                        <181>
3870 INPUT"GEBUEHREN LOESCHEN{3SPACE}N{3LE
     FT}";Q$                                   <049>
3880 IF Q$<>"J" AND Q$<>"N" THEN 3870          <188>
3890 IF Q$="J" THEN OPEN 1,8,2,"@:GEBUEHRE
     N,S,W":PRINT#1,0:CLOSE 1                  <165>
3900 OPEN 1,8,2,"GEBUEHREN,S,R"                <223>
3910 INPUT#1,GESB                              <087>
3920 CLOSE 1                                   <048>
3930 PRINT"{2DOWN}GESAMTBETRAG=";GESB;"DM"     <007>
3940 WAIT 198,255                              <111>
3950 GOTO 160                                  <157>
3960 :                                         <193>
3970 REM ** LO/HI-BERECHNUNG **                <049>
3980 POKE 832,SG-INT(SG/256)*256               <115>
3990 POKE 833,INT(SG/256)                      <240>
4000 GOSUB 4030                                <003>
4010 RETURN                                    <071>
4020 :                                         <253>
4030 REM ** SPEICHERGRENZE **                  <034>
4040 SG=PEEK(832)+PEEK(833)*256                <119>
4050 IF SG<53248 THEN BY=SG-15440              <246>
4060 IF SG>57343 THEN BY=SG-15440-4095         <123>
4070 PRINT:PRINT"BYTES BELEGT:";BY             <111>
4080 PRINT"BYTES FREI{2SPACE}:";46000-BY       <190>
4090 RETURN                                    <151>
4100 :                                         <078>
4110 REM ** FUNKTIONSTASTEN **                 <229>
4120 PRINT"{CLR,2DOWN,12SPACE}FUNKTIONSTAS
     TEN"                                      <202>
4130 PRINT"{12SPACE}YYYYYYYYYYYYYYYY{DOWN}"    <217>
4140 FOR FT=0 TO 7                             <020>
4150 PRINT" F";FT+1;":{2SPACE}";CHR$(34);      <071>
4160 SZ=54000+70*FT                            <255>
4170 E=USR(SZ):IF E=13 THEN 4210               <148>
4180 PRINT CHR$(E);                            <149>
4190 SZ=SZ+1                                   <085>
4200 GOTO 4170                                 <205>
4210 PRINT                                     <027>
4220 NEXT FT                                   <168>
4230 PRINT"{DOWN,12SPACE}AENDERN (J/N)?"       <088>
4240 GET Q$:IF Q$<>"J" AND Q$<>"N" THEN 42
     40                                        <109>
4250 IF Q$="N" THEN 160                        <044>
4260 PRINT"{HOME,4DOWN}"                       <232>
4270 FOR FT=0 TO 7                             <150>
4280 POKE 631,29:POKE 198,1                    <096>
4290 INPUT"{6RIGHT}";Q$                        <249>
4300 IF Q$="" THEN 4370                        <071>
4310 Q$=LEFT$(Q$,69)+CHR$(13)                  <092>
4320 SZ=54000+70*FT                            <159>
4330 FOR SP=1 TO LEN(Q$)                       <046>
4340 SYS 735,SZ,ASC(MID$(Q$,SP,1))             <025>
4350 SZ=SZ+1                                   <245>
4360 NEXT SP                                   <062>
4370 NEXT FT                                   <063>
4380 PRINT"{DOWN,12SPACE}SPEICHERN (J/N)?"     <147>
4390 GET Q$:IF Q$<>"J" AND Q$<>"N" THEN 43
     90                                        <010>
4400 IF Q$="N" THEN 160                        <195>
4410 OPEN 1,8,2,"@:F-TASTEN,S,W"               <011>
4420 FOR SZ=54000 TO 54559                     <222>
4430 PRINT#1,CHR$(USR(SZ));                    <092>
4440 NEXT SZ                                   <152>
4450 CLOSE 1                                   <068>
4460 GOTO 160                                  <157>
4470 :                                         <193>
4480 REM ** FUNKTIONSTASTEN SENDEN **          <021>
4490 D=ASC(B$)                                 <014>
4500 IF D=133 THEN FT=0                        <224>
4510 IF D=134 THEN FT=2                        <237>
4520 IF D=135 THEN FT=4                        <250>
4530 IF D=136 THEN FT=6                        <007>
4540 IF D=137 THEN FT=1                        <013>
4550 IF D=138 THEN FT=3                        <026>
4560 IF D=139 THEN FT=5                        <039>
4570 IF D=140 THEN FT=7                        <043>
4580 SZ=54000+70*FT                            <164>
4590 FOR W=1 TO 30:NEXT                        <125>
4600 SYS 62595                                 <178>
4610 B$=CHR$(USR(SZ))                          <249>
4620 GET#2,Q$                                  <181>
4630 IF DP THEN PRINT B$;                      <040>
4640 GOSUB 2100                                <098>
4650 SZ=SZ+1                                   <035>
4660 IF B$<>CHR$(13) THEN 4590                 <144>
4670 POKE 668,PEEK(667):GET#2,Q$               <062>
4680 RETURN                                    <232>
4690 :                                         <158>
4700 REM * CURSOR EIN *                        <011>
4710 E=PEEK(209)+PEEK(210)*256+PEEK(211)       <007>
4720 IF PEEK(E)<128 THEN POKE E,PEEK(E)+12
     8                                         <255>
4730 POKE 54272+E,0                            <114>
4740 RETURN                                    <036>
4750 :                                         <218>
4760 REM * CURSOR AUS *                        <084>
4770 E=PEEK(209)+PEEK(210)*256+PEEK(211)       <067>
4780 IF PEEK(E)>127 THEN POKE E,PEEK(E)-12
     8                                         <057>
4790 RETURN                                    <086>
4800 :                                         <012>
4810 REM ** DISKBEFEHLE **                     <041>
4820 PRINT"{CLR,2DOWN}'$' = DIRECTORY{2DOW
     N}"                                       <126>
4830 PRINT SPC(13) CHR$(34);:POKE 212,0:PR
     INT"{UP}"                                 <224>
4840 POKE 198,1:POKE 631,29                    <146>
4850 INPUT"DISKBEFEHL ";B$                     <191>
4860 IF B$="$" THEN PRINT:SYS 918:WAIT 198
     ,255:GOTO 160                             <099>
4870 OPEN 15,8,15                              <020>
4880 PRINT#15,B$                               <179>
4890 GOSUB 4950                                <140>
4900 CLOSE 15                                  <061>
4910 WAIT 198,255:GOTO 160                     <151>
4920 :                                         <133>
4930 REM ** DISKSTATUS **                      <155>
4940 OPEN 15,8,15                              <090>
4950 INPUT#15,B$,C$                            <120>
4960 PRINT"{2DOWN}DISKSTATUS: "B$;" ";C$;"
     {DOWN}"                                   <082>
4970 RETURN                                    <011>
4980 :                                         <193>
4990 REM * HELP *                              <157>
5000 PRINT                                     <052>
5010 PRINT"'C= S':DATEN SPEICHERN"             <106>
5020 PRINT"'C= E':SPEICHERN ENDE"              <150>
5030 PRINT"'C= A':DATEN AUSSENDEN"             <113>
5040 PRINT"'C= L':PUFFER LOESCHEN"             <141>
5050 PRINT"'C= G':GEBUEHREN"                   <014>
5060 PRINT"'C= B':BYTES FREI"                  <043>
5070 PRINT"'C= P':PRG-UEBERTRAGUNG"            <183>
5080 PRINT"'C= ←':OFF-LINE"                    <095>
5090 PRINT"'C= ↑':UNTERBRECHUNG"               <020>
5100 PRINT"'C= H':HELP"                        <213>
5110 RETURN                                    <151>
5120 :                                         <078>
5130 REM * HARDCOPY *                          <091>
5140 D=0:CLOSE 1:E=PEEK(840):OPEN 1,4,E        <020>
5150 FOR E=841 TO 845:PRINT#1,CHR$(PEEK(E)
     );:NEXT                                   <034>
5160 OPEN 2,3                                  <108>
5170 PRINT"{HOME}";                            <113>
5180 PRINT#1                                   <025>
5190 FOR Z=1 TO 25                             <035>
5200 FOR SP=1 TO 40                            <115>
5210 GET#2,E$:PRINT#1,E$;                      <204>
5220 NEXT SP                                   <157>
5230 NEXT Z                                    <094>
5240 PRINT#1:CLOSE 1:CLOSE 2                   <108>
5250 RETURN                                    <036>
```

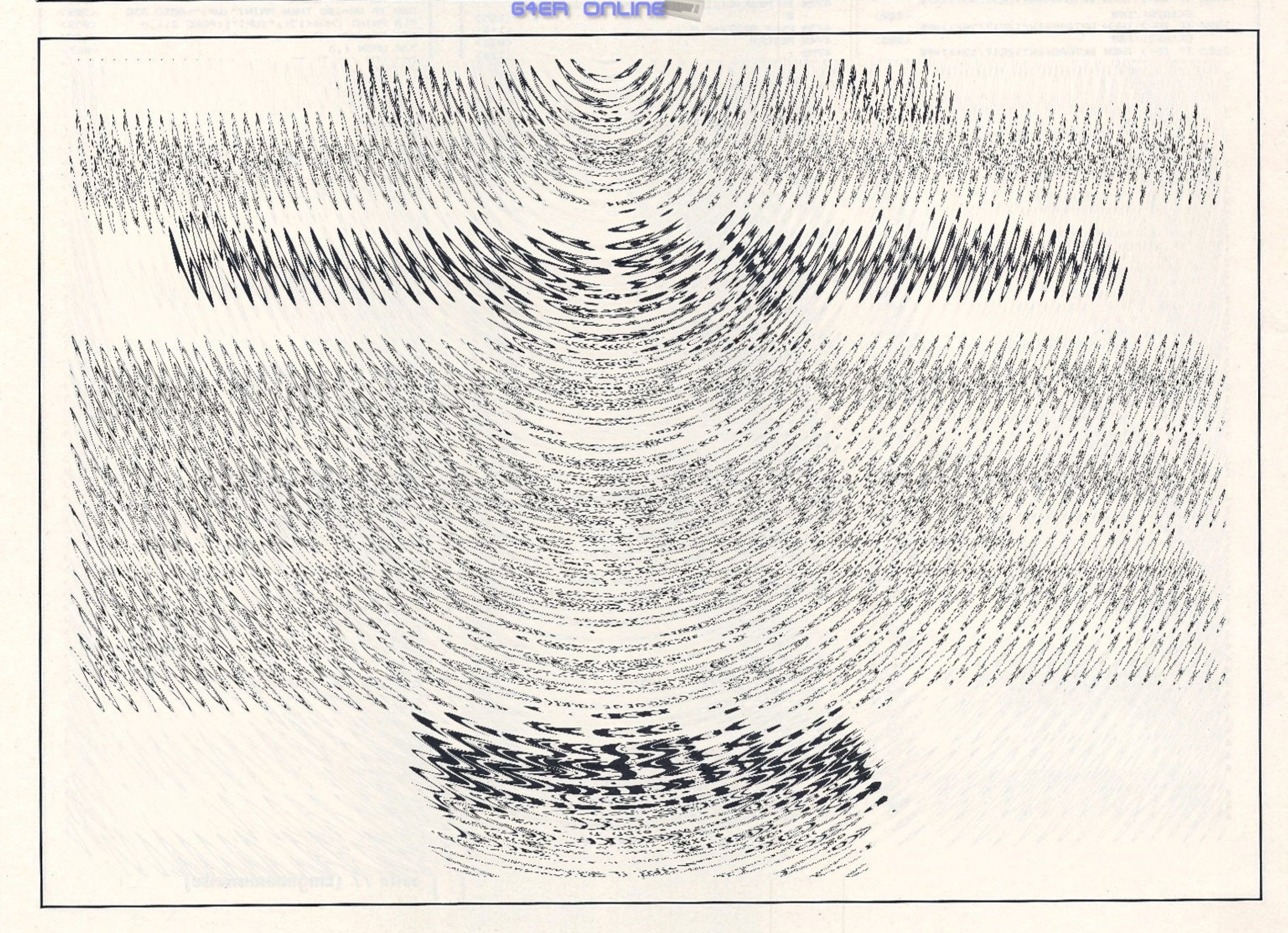
Listing »Terminal« (Schluß)

```
10 CLR                                         <166>
20 DIM ZE$(100)                                <063>
30 SG=PEEK(832)+PEEK(833)*256                  <190>
40 PRINT:PRINT"{BLACK,CLR,DOWN}"               <012>
50 PRINT"{8SPACE}PPPPPPPPPPPPPPPPPPPPPP"       <025>
60 PRINT"{8SPACE,RVSON,2SPACE}** EDITOR-ME
   NUE **{2SPACE}"                             <193>
70 PRINT"{8SPACE}YYYYYYYYYYYYYYYYYYYYYY{3D
   OWN}"                                       <016>
80 PRINT"{7SPACE}(1) - TEXT SCHREIBEN/EDIT
   IEREN{DOWN}"                                <013>
90 PRINT"{7SPACE}(2) - TEXT SPEICHERN{DOWN
   }"                                          <094>
100 PRINT"{7SPACE}(3) - TEXT LADEN{DOWN}"      <044>
110 PRINT"{7SPACE}(4) - TEXT AUSGEBEN{DOWN
    }"                                         <029>
120 PRINT"{7SPACE}(5) - DISKBEFEHLE{DOWN}"     <175>
130 PRINT"{7SPACE}(6) - PUFFER LOESCHEN{2D
    OWN}"                                      <206>
140 PRINT"{7SPACE}(7) - TERMINAL-PROGRAMM"     <012>
150 GET Q$:IF Q$="" THEN 150                   <025>
160 ON VAL(Q$) GOTO 210,1930,2140,1650,254
    0,2430,190                                 <110>
170 IF Q$="0" THEN GOSUB 2370:END              <138>
180 GOTO 150                                   <211>
190 GOSUB 2370:LOAD"TERMINAL",8                <232>
200 :                                          <002>
210 REM ** TEXT SCHREIBEN **                   <225>
220 PRINT"{CLR}":GOSUB 2300                    <216>
230 PRINT"{3DOWN,8SPACE}PUFFER LOESCHEN (J
    /N)?"                                      <230>
240 GET Q$:IF Q$<>"J" AND Q$<>"N" THEN 240     <137>
250 PRINT"{CLR}"                               <106>
260 IF Q$="J" THEN MAX=0:GOTO 320              <012>
270 GOSUB 1370:PRINT"{CLR}"                    <017>
280 FOR NR=0 TO 22                             <036>
290 PRINT ZE$(NR)                              <112>
300 NEXT NR                                    <079>
310 PRINT"{HOME}"                              <039>
320 NR=0                                       <195>
330 GOSUB 1030                                 <156>
340 GET Q$:IF Q$="" OR Q$="{HOME}" OR Q$="
    {CLR}" THEN 340                            <181>
350 IF Q$="←" THEN MAX=MAX-1:GOSUB 1490:GO
    TO 40                                      <113>
360 GOSUB 1080                                 <191>
370 IF ASC(Q$)>31 AND ASC(Q$)<128 THEN 460     <105>
380 IF (Q$="{LEFT}" OR Q$=CHR$(20)) AND PO
    S(0)=0 THEN 330                            <186>
390 IF Q$="{RIGHT}" AND POS(0)=38 THEN 330     <221>
400 IF Q$="{UP}" THEN GOSUB 1220:GOTO 330      <106>
410 IF Q$="{DOWN}" THEN GOSUB 1300:GOTO 33
    0                                          <243>
420 IF Q$=CHR$(13) OR Q$=CHR$(141) THEN 49
    0                                          <156>
430 IF Q$="{INST}" AND PEEK(1024+38+PEEK(2
    14)*40)<>32 THEN 330                       <065>
440 IF Q$="{CTRL-I}" AND POS(0)=0 THEN GOS
    UB 680:GOTO 330                            <174>
450 IF Q$="{CTRL-D}" AND POS(0)=0 THEN GOS
    UB 860:GOTO 330                            <179>
460 PRINT Q$;:POKE 212,0                       <216>
470 IF POS(0)<39 THEN 330                      <248>
480 :                                          <027>
490 REM * EINLESEN *                           <033>
500 IF NR=100 THEN PRINT"{UP}":GOTO 330        <209>
510 PRINT CHR$(13);"{UP}";:POKE 211,0          <032>
520 ZE$(NR)=""                                 <180>
530 OPEN 1,3                                   <067>
540 FOR X=1 TO 39                              <234>
550 GET#1,E$                                   <178>
560 ZE$(NR)=ZE$(NR)+E$                         <095>
570 NEXT X                                     <022>
580 CLOSE 1                                    <023>
590 N=39                                       <188>
600 IF MID$(ZE$(NR),N,1)<>" " THEN 620         <114>
610 N=N-1:IF N>0 THEN 600                      <191>
620 ZE$(NR)=LEFT$(ZE$(NR),N)                   <027>
630 NR=NR+1:IF NR>MAX THEN MAX=NR              <032>
640 PRINT                                      <027>
650 IF NR=100 THEN POKE 53280,2                <152>
660 GOTO 330                                   <181>
670 :                                          <218>
680 REM * INSERT *                             <098>
690 IF MAX=100 THEN RETURN                     <157>
700 Z=PEEK(214):N=NR                           <078>
710 FOR X=PEEK(214) TO 23                      <006>
720 PRINT"{39SPACE}";CHR$(20);" ":NEXT X       <247>
730 POKE 214,Z+1:PRINT"{UP}"                   <019>
740 E=NR+22-Z:IF E>MAX THEN E=MAX              <181>
750 FOR N=NR TO E                              <250>
760 PRINT ZE$(N)                               <245>
770 NEXT N                                     <213>
780 POKE 214,Z:PRINT"{UP}"                     <107>
790 FOR Z=MAX TO NR STEP-1                     <085>
800 ZE$(Z+1)=ZE$(Z)                            <108>
810 NEXT Z                                     <009>
820 MAX=MAX+1                                  <144>
830 ZE$(NR)=""                                 <235>
840 RETURN                                     <217>
850 :                                          <143>
860 REM * DELETE *                             <245>
870 IF MAX=0 THEN RETURN                       <241>
880 Z=PEEK(214)                                <041>
890 FOR X=PEEK(214) TO 23                      <187>
900 PRINT"{39SPACE}";CHR$(20);" ":NEXT X       <172>
910 POKE 214,Z:PRINT"{UP}"                     <237>
920 E=NR+24-Z:IF E>MAX THEN E=MAX              <108>
930 FOR N=NR+1 TO E                            <138>
940 PRINT ZE$(N)                               <170>
950 NEXT N                                     <137>
960 POKE 214,Z:PRINT"{UP}"                     <031>
970 MAX=MAX-1                                  <039>
980 FOR Z=NR TO MAX                            <142>
990 ZE$(Z)=ZE$(Z+1)                            <074>
1000 NEXT Z                                    <199>
1010 RETURN                                    <131>
1020 :                                         <057>
1030 REM * CURSOR EIN *                        <167>
1040 P=PEEK(209)+PEEK(210)*256+PEEK(211)       <174>
1050 IF PEEK(P)<128 THEN POKE P,PEEK(P)+12
     8:POKE 54272+P,0                          <231>
1060 RETURN                                    <182>
1070 :                                         <108>
1080 REM * CURSOR AUS *                        <230>
1090 P=PEEK(209)+PEEK(210)*256+PEEK(211)       <224>
1100 IF PEEK(P)>127 THEN POKE P,PEEK(P)-12
     8                                         <236>
1110 RETURN                                    <232>
1120 :                                         <158>
1130 REM * ZURUECKBLAETTERN *                  <027>
1140 PRINT"{CLR}"                              <232>
```

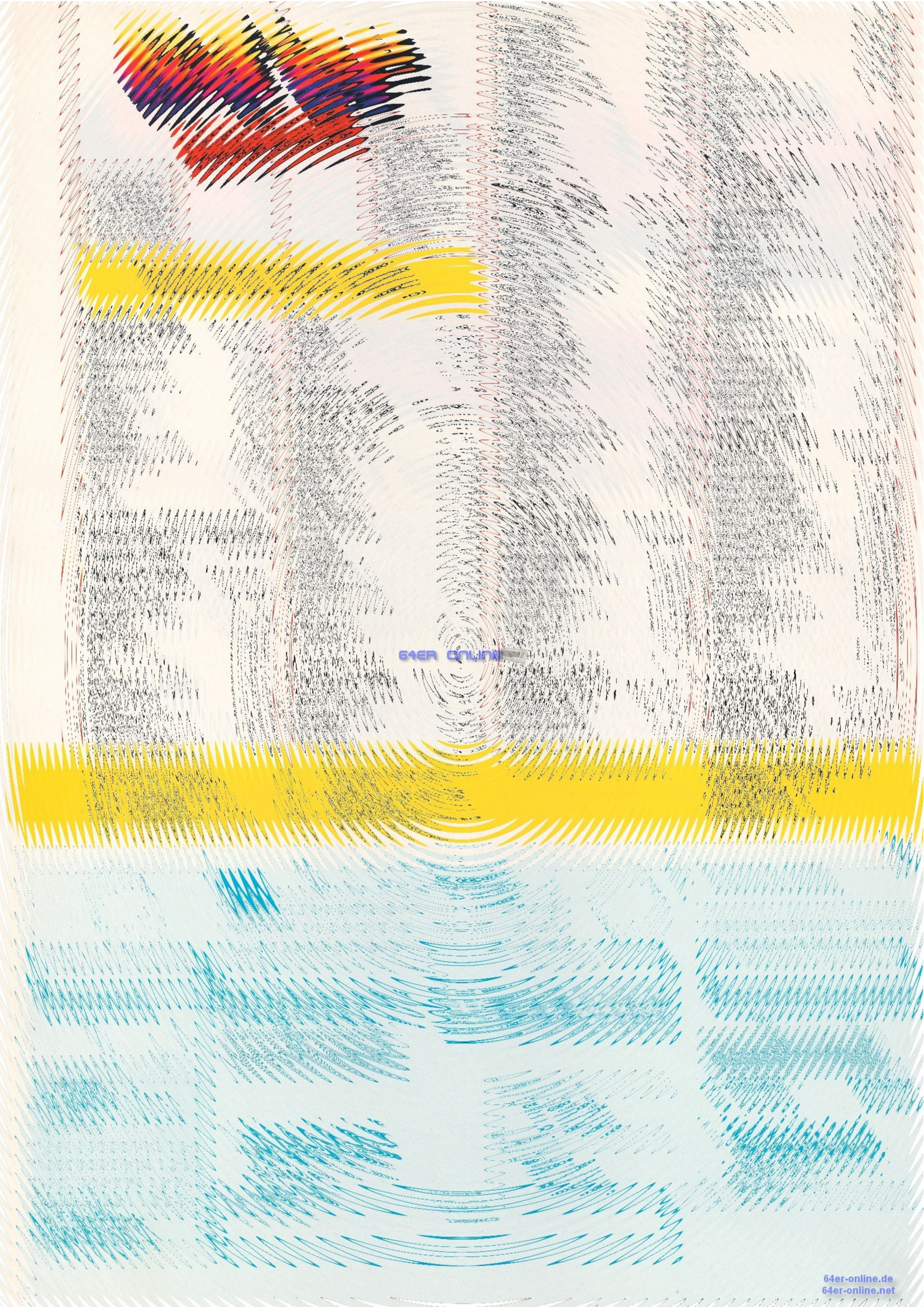
Listing »Editor«. Bitte beachten Sie Seite 77 (Eingabehinweise)

```
1150 NR=NR-22:IF NR<0 THEN NR=0                <244>
1160 FOR Z=NR TO NR+22                         <011>
1170 PRINT ZE$(Z)                              <157>
1180 NEXT Z                                    <124>
1190 PRINT"{HOME}"                             <154>
1200 RETURN                                    <066>
1210 :                                         <248>
1220 REM * CURSOR UP *                         <046>
1230 IF NR=100 THEN POKE 53280,PEEK(838)       <098>
1240 IF PEEK(214)=1 AND NR>0 THEN GOSUB 11
     30:RETURN                                 <229>
1250 IF PEEK(214)=1 THEN RETURN                <051>
1260 PRINT"{UP}";                              <153>
1270 NR=NR-1                                   <200>
1280 RETURN                                    <147>
1290 :                                         <073>
1300 REM * CURSOR DOWN *                       <018>
1310 IF NR=MAX THEN RETURN                     <027>
1320 IF PEEK(214)>23 THEN PRINT ZE$(NR):NR
     =NR+1:RETURN                              <059>
1330 PRINT"{DOWN}";                            <096>
1340 NR=NR+1                                   <014>
1350 RETURN                                    <217>
1360 :                                         <143>
1370 REM * SPEICHER-->VARIABLE *               <115>
1380 PRINT"{CLR,5DOWN,9SPACE}EINEN MOMENT
     BITTE..."                                 <111>
1390 NR=0:ZE$(0)=""                            <105>
1400 FOR SZ=15440 TO SG                        <153>
1410 E=USR(SZ)                                 <051>
1420 IF E=13 THEN NR=NR+1:ZE$(NR)="":GOTO
     1450                                      <092>
1430 IF LEN(ZE$(NR))=39 THEN NR=NR+1:ZE$(N
     R)=""                                     <100>
1440 ZE$(NR)=ZE$(NR)+CHR$(E)                   <198>
1450 IF NR<100 THEN NEXT SZ                    <244>
1460 IF NR=100 THEN PRINT"{4DOWN,8SPACE}AC
     HTUNG, TEXT ZU LANG !!!":WAIT 198,255     <212>
1470 MAX=NR:POKE 198,0:RETURN                  <146>
1480 :                                         <007>
1490 REM * VARIABLE-->SPEICHER *               <235>
1500 POKE 53280,PEEK(838)                      <092>
1510 PRINT"{CLR,5DOWN,9SPACE}EINEN MOMENT
     BITTE...{2DOWN}"                          <019>
1520 SG=15440                                  <063>
1530 FOR NR=0 TO MAX                           <140>
1540 IF ZE$(NR)="" THEN 1590                   <181>
1550 FOR Z=1 TO LEN(ZE$(NR))                   <062>
1560 Q$=MID$(ZE$(NR),Z,1)                      <247>
1570 POKE SG,ASC(Q$):SG=SG+1                   <012>
1580 NEXT Z:ZE$(NR)=""                         <242>
1590 POKE SG,13:SG=SG+1                        <248>
1600 NEXT NR                                   <104>
1610 SG=SG-1                                   <018>
1620 GOSUB 2300:WAIT 198,255                   <226>
1630 RETURN                                    <242>
1640 :                                         <168>
1650 REM ** DATEN AUSGEBEN **                  <101>
1660 PRINT"{CLR}"                              <242>
1670 GOSUB 2300                                <222>
1680 IF SG=15440 THEN 40                       <118>
1690 PRINT"{DOWN}'F1' = BILDSCHIRM"            <113>
1700 PRINT"'F3' = DRUCKER"                     <155>
1710 PRINT"'F5' = HARDCOPY{DOWN}"              <002>
1720 WAIT 198,255                              <186>
1730 D=0                                       <238>
1740 FOR SZ=15440 TO SG                        <238>
1750 IF PEEK(198)=0 THEN 1830                  <113>
1760 GET B$                                    <237>
1770 IF B$=" " THEN WAIT 198,255:GET B$        <187>
1780 IF B$="←" THEN POKE 53280,6:CLOSE 1:G
     OTO 40                                    <020>
1790 IF B$="{F3}" AND D=0 THEN Z=PEEK(840)
     :POKE 53280,2                             <217>
1800 IF B$="{F3}"AND D=0 THEN OPEN 1,4,Z:F
     OR Z=841 TO 845:PRINT#1,CHR$(PEEK(Z))
     ;:NEXT:D=1                                <099>
1810 IF B$="{F1}" THEN D=0:CLOSE 1:POKE 53
     280,PEEK(838)                             <018>
1820 IF B$="{F5}" THEN GOSUB 2720              <144>
1830 C$=CHR$(USR(SZ))                          <019>
1840 PRINT C$;                                 <114>
1850 IF D THEN PRINT#1,C$;                     <078>
1860 IF SZ=53247 THEN SZ=57343                 <070>
1870 NEXT SZ                                   <132>
1880 CLOSE 1:POKE 53280,PEEK(838)              <229>
1890 GET B$:IF B$="" THEN 1890                 <010>
1900 IF B$="{F5}" THEN GOSUB 2720              <224>
1910 GOTO 40                                   <106>
1920 :                                         <193>
1930 REM ** DATEN SPEICHERN **                 <213>
1940 PRINT"{CLR}"                              <011>
1950 GOSUB 2300                                <247>
1960 IF SG=15440 THEN 40                       <143>
1970 DN$=""                                    <101>
1980 INPUT"{2DOWN}DATEINAME ";DN$              <167>
1990 IF DN$="" THEN 40                         <015>
2000 OPEN 1,8,2,DN$+",S,W"                     <059>
2010 GOSUB 2660                                <060>
2020 PRINT#1,CHR$(USR(15440));                 <057>
2030 IF ST<>0 THEN 2100                        <037>
2040 CMD 1                                     <205>
2050 FOR SZ=15441 TO SG                        <039>
2060 PRINT CHR$(USR(SZ));                      <181>
2070 IF SZ=53247 THEN SZ=57343                 <025>
2080 NEXT SZ                                   <055>
2090 PRINT#1,"";                               <166>
2100 CLOSE 1:CLOSE 15                          <077>
2110 IF VAL(B$)<>0 THEN WAIT 198,255           <132>
2120 GOTO 40                                   <061>
2130 :                                         <148>
2140 REM *** DATEN LADEN ***                   <191>
2150 DN$=""                                    <026>
2160 INPUT"{CLR,3DOWN}DATEINAME ";DN$          <000>
2170 IF DN$="" THEN 40                         <196>
2180 SG=15440                                  <214>
2190 OPEN 1,8,2,DN$+",S,R"                     <245>
2200 GOSUB 2660                                <251>
2210 IF VAL(B$)<>0 THEN 2250                   <181>
2220 POKE 144,0:POKE 832,80:POKE 833,60        <067>
2230 SYS 848:REM LOAD-ROUTINE                  <060>
2240 SG=PEEK(832)+PEEK(833)*256-1              <068>
2250 CLOSE 1:CLOSE 15                          <227>
2260 IF SG=-1 THEN PRINT"{2DOWN}ACHTUNG, P
     UFFER VOLL !!!{2DOWN}":SG=65535           <131>
2270 GOSUB 2300:WAIT 198,255                   <110>
2280 GOTO 40                                   <221>
2290 :                                         <052>
2300 REM ** SPEICHERGRENZE **                  <089>
2310 IF SG<53248 THEN BY=SG-15440              <036>
2320 IF SG>57343 THEN BY=SG-15440-4095         <169>
2330 PRINT:PRINT"BYTES BELEGT:";BY             <157>
2340 PRINT"BYTES FREI{2SPACE}:";46000-BY       <236>
2350 RETURN                                    <197>
2360 :                                         <123>
2370 REM ** LO/HI-BERECHNUNG **                <235>
2380 LO=SG-INT(SG/256)*256                     <026>
2390 HI=INT(SG/256)                            <140>
2400 POKE 832,LO:POKE 833,HI                   <144>
2410 RETURN                                    <001>
2420 :                                         <183>
2430 REM ** PUFFER LOESCHEN **                 <215>
2440 PRINT"{CLR,2DOWN}"                        <035>
2450 INPUT"PUFFER LOESCHEN (J/N)";Q$           <197>
2460 IF Q$<>"J" AND Q$<>"N" THEN 2450          <035>
2470 PRINT"{DOWN}"                             <157>
2480 IF Q$="N" THEN GOSUB 2300:WAIT 198,25
     5:GOTO 40                                 <083>
2490 SG=15440                                  <013>
2500 GOSUB 2300                                <031>
2510 WAIT 198,255                              <211>
2520 GOTO 40                                   <206>
2530 :                                         <037>
2540 REM ** DISKBEFEHLE **                     <066>
2550 PRINT"{CLR,2DOWN}'$' = DIRECTORY{2DOW
     N}"                                       <151>
2560 PRINT SPC(13) CHR$(34);:POKE 212,0:PR
     INT"{UP}"                                 <250>
2570 POKE 198,1:POKE 631,29                    <172>
2580 INPUT"DISKBEFEHL ";B$                     <217>
2590 IF B$="$" THEN PRINT:SYS 918:WAIT 198
     ,255:GOTO 40                              <074>
2600 OPEN 15,8,15                              <045>
2610 PRINT#15,B$                               <204>
2620 GOSUB 2680                                <163>
2630 CLOSE 15                                  <086>
2640 WAIT 198,255:GOTO 40                      <125>
2650 :                                         <158>
2660 REM ** DISKSTATUS **                      <180>
2670 OPEN 15,8,15                              <115>
2680 INPUT#15,B$,C$                            <145>
2690 PRINT"{2DOWN}DISKSTATUS: "B$;" ";C$;"
     {DOWN}"                                   <107>
2700 RETURN                                    <036>
2710 :                                         <218>
2720 REM * HARDCOPY *                          <231>
2730 D=0:CLOSE 1:Z=PEEK(840):OPEN 1,4,Z        <202>
2740 FOR Z=841 TO 845:PRINT#1,CHR$(PEEK(Z)
     );:NEXT                                   <216>
2750 OPEN 2,3                                  <248>
2760 PRINT"{HOME}";                            <253>
2770 PRINT#1                                   <165>
2780 FOR Z=1 TO 25                             <175>
2790 FOR SP=1 TO 40                            <255>
2800 GET#2,E$:PRINT#1,E$;                      <088>
2810 NEXT SP                                   <041>
2820 NEXT Z                                    <235>
2830 PRINT#1:CLOSE 1:CLOSE 2                   <249>
2840 RETURN                                    <177>
```

Listing »Editor« (Schluß)






64ER ONLINE

Commodore-Sportservice — Heimcomputer zur Turnierauswertung

Eine handvoll Unternehmen hat sich auf das Auswerten von Sportveranstaltungen spezialisiert. So auch der Commodore-Sportservice, der bei Veranstaltungen von Rang und Namen zu finden ist. Diesmal war er beim Berliner Springreiten, bei dem Conrad Homfeld den World Cup mit nach Hause nahm.

Sie kennen bestimmt die Spielstand-Einblendungen bei Sportübertragungen am unteren Rand Ihres Fernsehers. Was dahinter steckt, ist ein sehr großer Aufwand an Elektronik. Aber keine Riesen-Computer, sondern viele Kleine machen die Arbeit. Durch die Verwendung von Mikrocomputern wird das System so flexibel, daß jedes anfallende Problem damit gemeistert werden kann. Es ist ein großer Unterschied zwischen einer Hallenveranstaltung und einem Skirennen, bei dem Minusgrade nicht nur den Computern zu schaffen machen und die Anlage auf Pistenraupen durchgeschüttelt wird. Ebenso wechseln die Anforderungen an die Leistungsfähigkeit des Systems von Fall zu Fall.

Ein anderer Grund, der für ein System aus mehreren kleinen Computern spricht, ist die Fehlersuche. Denn ein großes System ist wesentlich servicefreundlicher, wenn es aus kleinen Einheiten zusammengesetzt ist. Stellen Sie sich vor, Sie hätten eine große EDV-Anlage bei einer Veranstaltung im Keller stehen und nur Terminals am Richterturm. Nun fällt ein Bauteil aus. Die anschließende Fehlersuche wird sich dann als sehr schwierig erweisen, da der Fehler erst lokalisiert werden muß. Außerdem liegt während der Reparaturzeit das System lahm und die Veranstaltung müßte unterbrochen werden. Im Gegensatz dazu lassen sich Fehler in einem System, das aus mehreren kleinen Computern zusammengesetzt ist, sehr schnell beheben. Man sieht schnell, welcher Computer defekt ist und tauscht ihn einfach gegen einen neuen aus. Dadurch kann ein Fehler so schnell behoben werden, daß der Außenstehende vielleicht gar nichts davon merkt.

Das System des Commodore-Sportservice besteht aus:
30 SX 64, der tragbaren C 64-Version mit integriertem Diskettenlaufwerk
25 CBM 8296 Personal Computer



Hier laufen die Signale der Schriftgeneratoren zusammen. Auf dem linken Monitor die Rangliste, auf dem rechten das World Cup-Logo. Das Bild auf dem mittleren Monitor bekommt der Ü-Wagen. Die Schrift wird mit dem »Blue Box-Trick« ins Fernsehbild eingeblendet.

25 CBM 4040 Floppy-Disk-Doppellaufwerken
25 MPP 1361 Drucker mit IEEE-Anschluß
2 Festplattenlaufwerke 20 MByte
8 Telefonmodems zur Fernübertragung von Daten
4 Videogeneratoren zur Fernsehdirekteinblendung
50 1701 Farbmonitore
6000 W Notstromversorgung, falls der Strom bis zu fünf Minuten ausbleiben sollte.

Mit diesen Geräten hat der Commodore-Sportservice, kurz COSS genannt, seit 1982 über 100 nationale und internationale Sportveranstaltungen im europäischen Raum betreut und Veranstalter, Zuschauer, Presse und Fernsehen mit Informationen über den Wettkampf beliefert. Wenn beispielsweise Journalisten wissen möchten, wieviel ein Teilnehmer während des Wettkampfes schon an Prämien »kassiert« hat, können sie sich beim Sportservice jederzeit eine Gewinnliste holen.

Die SX 64-Computer werden vor allem bei Sportarten eingesetzt, die nach vielfältigen Kriterien beurteilt werden. Ein sehr gutes Beispiel dafür sind die nordischen Skimeisterschaften, wo verschiedene Disziplinen gleichzeitig ausgetragen werden. Es müssen dort viele Ranglisten gleichzeitig geführt werden: Skispringen, Langlauf, Biathlon, etc. Jeder SX 64 hat dazu ein Programm, das eine bestimmte Auswertung übernimmt.

Den Commodore-Sportservice gibt es seit Januar 1981, wo sich das Team mit der Betreuung des FIS-Slaloms der Herren in Lermoos profilieren konnte. Die siebenköpfige Mannschaft besteht aus Programmierern und Elektronikern, die auch die gesamte Software und Interface-Hardware entwickelt haben. Bei großen Vorhaben wird das Team entsprechend verstärkt, so daß auch schon mal 15 Mann Kabel hinter Deckenverkleidungen verlegen.

Inzwischen hat man bei Commodore Erfahrung mit alpinen und nordischen Skimeisterschaften, im Kanu-, Rad-, Reit- und Rudersport, bei Schwimmveranstaltungen und im Turmspringen. Bisher war die Betreuung von Veranstaltungen immer ein Erfolg für den Veranstalter und den COSS. Auch die Fernsehanstalten stellen dem Commodore-Sportservice ein sehr gutes Zeugnis aus.

Es werden aber nicht nur Zeitauswertungen gemacht. Die Computer nehmen dem Veranstalter auch viel organisatorische Arbeit ab. Zum Beispiel bei der Erstellung der Melde- und Startlisten. Die Melde- und Startlisten von Hand nach Klassen zu ordnen und mit Schreibmaschine zu tippen, kostet manuell sehr viel mehr Zeit.

Besonders kompliziert wird der Vorgang, wenn ein Reiter mit mehreren Pferden an einem Wettbewerb teilnehmen möchte. Der Reiter kann nämlich selbst bestimmen, mit welchen Pferden in welcher Reihenfolge er springen will. Das Reglement schreibt aber vor, daß zwischen zwei Durchgängen eines Reiters mindestens sechs andere Teilnehmer auf dem Parcour gewesen sein müssen. Bestehen nur vier oder fünf Reiter auf die selbst festgelegte Startreihenfolge, kommt die Meldestelle ins »Rotieren«.

Die Startliste ist aber nicht nur der einzige »Zusatzservice«. Ist eine Prüfung beendet, kann sofort eine Liste mit den Zwischenergebnissen gedruckt, vervielfältigt und verteilt werden. Diese Liste zeigt zugleich die Startreihenfolge der nächsten Prüfung.

Bis zu 30 SX 64 in einem Informationssystem

Wie schon erwähnt, werden die SX 64 zum Erstellen von Punktelisten, Ranglisten, Startreihenfolgen verwandt, wenn eine große Informationsflut zu bewältigen ist. So liefert, etwa bei nordischen Skimeisterschaften, einer die augenblickliche Rangfolge im Sprunglauf, ein anderer die Langlaufergebnisse oder den Stand in der nordischen Kombination. Auch eine Aufstellung ausgeschiedener Teilnehmer der einzelnen Klassen kann erstellt werden.

Je nach Wichtigkeit wird der Videoausgang eines bestimmten SX 64 einem Videoverstärker (zur Ansteuerung von bis zu 50 Monitoren) zugeschaltet und das verstärkte Bild dem Infosystem zugeführt. Über dieses System werden Veranstalter, Teilnehmer und Zuschauer vom COSS mit aktuellen Informationen versorgt. Dazu werden, wo sich die Möglichkeit bietet, Farbmonitore aufgestellt. In Hallen oder in Korridoren (falls die Veranstaltung innerhalb eines Gebäudes stattfindet), in Sprecherkabinen von Rundfunkreportern und in Pressesälen. Über das Infosystem wird meist das momentane Geschehen mit einer aktuellen Auswertung gezeigt.

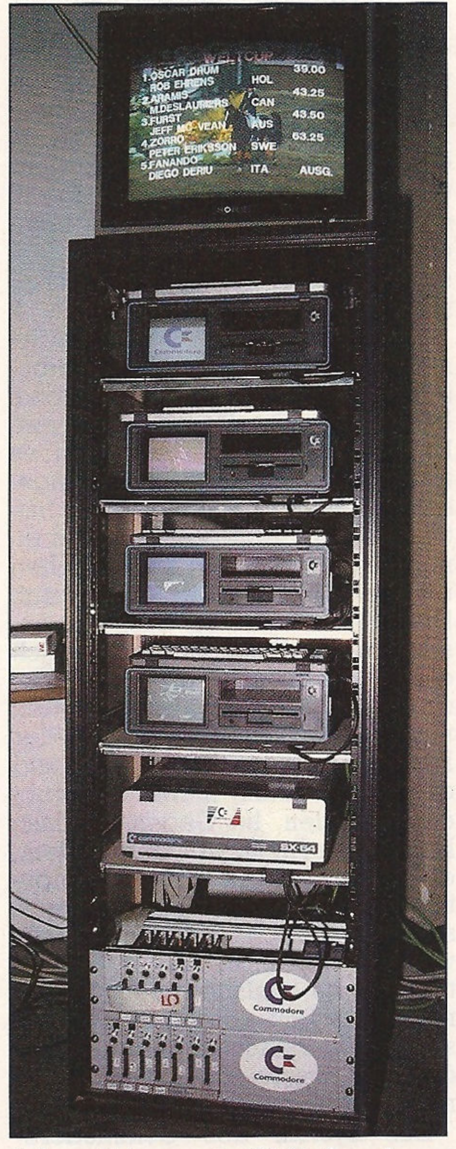
Über ein selbstentwickeltes Interface werden die SX 64 miteinander »vernetzt«, meist in Gruppen von fünf Computern. Die fünf Heimcomputer werden mitsamt dem Interface in einfach zu transportierenden HiFi-Turm-artigen Schaltschränken untergebracht. Durch dieses vom COSS entwickelte Interface können die SX 64 auf eine der 20 MByte-Festplatten zugreifen und von dort ständig die neuesten Ergebnisse holen und auswerten. Es ist auch möglich, vom Richterturm aus, mit einem CBM 8296, den SX-64 Kommandos zu geben.

Das selbstentworfene Interface empfängt über eine V.24-Schnittstelle von einem CBM 8296 oder von Festplatte Daten und sendet sie über den User-Port an einen SX 64. Am User-Port wird softwareseitig eine 8-Bit-Parallelschnittstelle simuliert, die der Centronics-Norm recht ähnlich ist. Anhand einer Geräteadresse erkennt das Interface, an welchen SX 64 die Daten geleitet werden sollen. Die Geräteauswahl findet also im Interface statt, und nicht im SX 64.

Drei Tage Aufbau

Der technische Aufwand läßt erahnen, welche Arbeit investiert werden muß, um die gesamte Anlage aufzubauen. Beim World Cup der Springreiter in Berlin mußten dazu innerhalb drei Tagen etliche hundert Meter Kabel verlegt werden, um aus den einzelnen Computern ein einziges System zu machen und um die Verbindung zum Übertragungswagen herzustellen.

Das Herz des Systems befindet sich mit auf dem Richterturm, hinter der Richterbank. In der Berliner Deutschlandhalle standen dort vier CBM 8296 mit Diskettenlaufwerken und Druckern und zwei Schriftgeneratoren (Aston 3). Anstelle der 8296 könnten genauso C 64 verwendet werden, so der COSS-Geschäftsführer Dr. Hehenwarter, aber die 8296 sind nun mal mit entsprechenden Programmen vorhanden. Mit drei der vier CBM 8296 wurde über Schriftgeneratoren das Bild fürs Fernsehen und das Infosystem erzeugt. Für das Fernsehbild hatte man zwei Computer: Einen für die Zeit- und Fehlerpunkte-Einspielung und einen für Rangliste. Dadurch konnten beiden Auswertungen abwechselnd dem Fernsehbild zugemischt werden. So lange ein Reiter sich auf dem Parcour befand, wurden in der Regel die Zeit und die Fehlerpunkte eingeblendet. Ansonsten zeigte man den World Cup-Stand. Über das Infosystem wurde der aktuelle Ritt mit dem Zeitstand gezeigt. War einmal kein Reiter auf dem Parcour, konnten von den SX 64 Logos oder Pictogramme abgerufen und übers Infosystem gezeigt werden. Das war beispielsweise ein stilisierter Reiter oder das Commodore-Firmenzeichen. Mit dem vier-



Mehrere Heimcomputer SX 64 übernehmen Auswertungen für das Infosystem. Zuschauer und Teilnehmer erfahren, was aktuell ist.



Sehr viele Kabel sind zu verlegen. Auf dem Foto sehen Sie den kleinen Ausschnitt der Anlage auf dem Richterturm.



Im Keller befindet sich unscheinbar das Gedächtnis des ganzen COSS-Systems: zwei Festplattenlaufwerke mit insgesamt 40 MByte.

ten der CBM 8296 wurde die Anzeigetafel in der Halle angesteuert.

Dem Infosystem waren in der Deutschlandhalle etwa 50 Monitore in den Gängen und im Pressesaal angeschlossen, wo das laufende Geschehen gezeigt wurde. Ins Kamerabild wurden ständig die aktuellen Zwischenergebnisse eingeblendet. Das laufende Geschehen nahm der Commodore-Sportservice mit einer eigenen Kamera auf.

In Berlin wurden die gemessenen Zeiten ausnahmsweise auf Zuruf eingegeben. Im Regelfall ist das nicht notwendig. Ein CBM 8296 wird dazu über eine V.24-Schnittstelle direkt mit der Elektronik der Zeitmessung verbunden. Vor dem Wettkampf werden die Nennungen eingegeben. Die Nennung umfaßt neben den Namen, mitgebrachte Punkte für die World Cup-Wertung, die Nation, die Klasse und den Verein. Aus diesen Daten wird eine Startliste für den Veranstalter erstellt. Was während des Turnieres noch getan werden muß, beschränkt sich auf die Eingabe der Fehlerpunkte.

Die Zeitnahme ist in diesem Zusammenhang auch das einzige, das der Commodore-Sportservice nicht selbst macht. Das übernehmen Spezialfirmen wie Omega, Longines und Alge, die Ihnen bestimmt als Uhrenhersteller bekannt sind. Diese Firmen erledigen die technische Zeiterfassung und stellen das Zeitsignal den Commodore-Leuten bereit. Der Vorteil dieser Direktverbindung liegt auf der Hand: keine Übertragungsfehler mehr beim Zurufen der Zeit und keine Eingabefehler mehr.

Zuschauer verlangen gute Informationsdarstellungen

Damit die Zuschauer zu Hause vor dem Fernseher und die Zuschauer vor Ort gut über das Geschehen informiert werden können, sind viele Probleme zu lösen und hohe Investitionen nötig.

Ein Problem, das häufig auftrete, sei der Anschluß der Score-Boards, der großen Anzeigetafeln, die in jedem Stadion oder jeder Sporthalle zu finden sind, so Dr. Hehenwarter. Der Anschluß der Tafeln erfolgt zwar über RS232-Schnittstellen, die aber teilweise alles andere als genormte Baudraten haben.

Sehr hohe Investitionen waren für die zwei Schriftgeneratoren nötig. Ohne Zubehör wie Einstellmonitore, Floppy-Disk-Laufwerk kostet ein einziger Schriftgenerator (Aston 3) über 100 000 Mark. Die Schriftgeneratoren erzeugen die Schrift, wie sie später im Fernsehen oder auf den Monitoren des Info-Systems zu sehen ist. Von den beiden Schriftgeneratoren erzeugt einer die Schrifteinblendungen für das Info-System, der andere liefert das Videosignal der Schrift dem Übertragungswagen. Diese Schriftgeneratoren können fast alles erdenkliche, was mit Schrift zu tun hat: verschiedene Schrifttypen, Farbwechsel, Blocksatz, verkleinern, vergrößern, verschieben oder schrägstellen der Schrift und mehr. Der Anzahl der Schrifttypen sind keine Grenzen gesetzt. Die einzelnen Typen können mit einer Kamera aufgenommen und durch den Schriftgenerator digitalisiert werden. Die Zeichensätze werden dann auf Diskette gespeichert und können jederzeit geladen werden.

Die Ansteuerung der Schriftgeneratoren durch den Computer erfolgt denkbar einfach: genau wie ein Drucker. Die Texte werden einfach im ASCII-Code an den Schriftgenerator gesandt und können dort bearbeitet werden. Die Generatoren haben auch eine eigene Tastatur und ein einfaches Textverarbeitungsprogramm zur direkten Texteingabe.

»Früher«, so Dr. Hehenwarter, »genügte das Schriftbild eines Computers noch den Qualitätsansprüchen. Heutezutage ist es aber undenkbar, eine Matrixschrift mit vielleicht 8x8 Punkten ins Fernsehbild einzublenden.«

90 Programme für ein Reitturnier

Neunzig Programme wurden in Berlin zur Auswertung des World Cup-Turniers gebraucht. Alle hat das COSS-Team selbst geschrieben. »Das geht auch nicht anders«, so ein Mitarbeiter, »denn die Programme müssen häufig noch am Richterturm umgeschrieben werden.« Es ist also sehr wichtig, daß mindestens ein Programmierer, der an der Softwareerstellung beteiligt war und deshalb das Programm sehr gut kennt, mit von der Partie ist. Man braucht sich dazu nur ein Programm näher betrachten. Man wird schnell feststellen, daß es professionelle Programme sind, die mit den meisten handelsüblichen Dateiverwaltungen nur den Namen gemeinsam haben. Hier, beim Schreiben, Verän-

dern, Anpassen und Compilieren der Programme werden auch die 4040-Laufwerke benötigt.

SX 64 – Garbage Collection in einer Sekunde

Ein Ansporn für C 64-Programmierer dürfte die Tatsache sein, daß die COSS-Leute für den SX 64 eine Garbage Collection geschrieben haben, die maximal eine Sekunde braucht, um den gesamten Stringspeicher des SX 64 oder C 64 »aufzuräumen«. Jeder SX 64/C 64-Programmierer hat wahrscheinlich schon mit dem Problem der Garbage Collection, die bis zu einer halben Stunde dauern kann, gekämpft. Ohne diese schnelle Garbage Collection-Routine wäre der Einsatz des SX 64 als Computer zur schnellen Auswertung gar nicht denkbar.

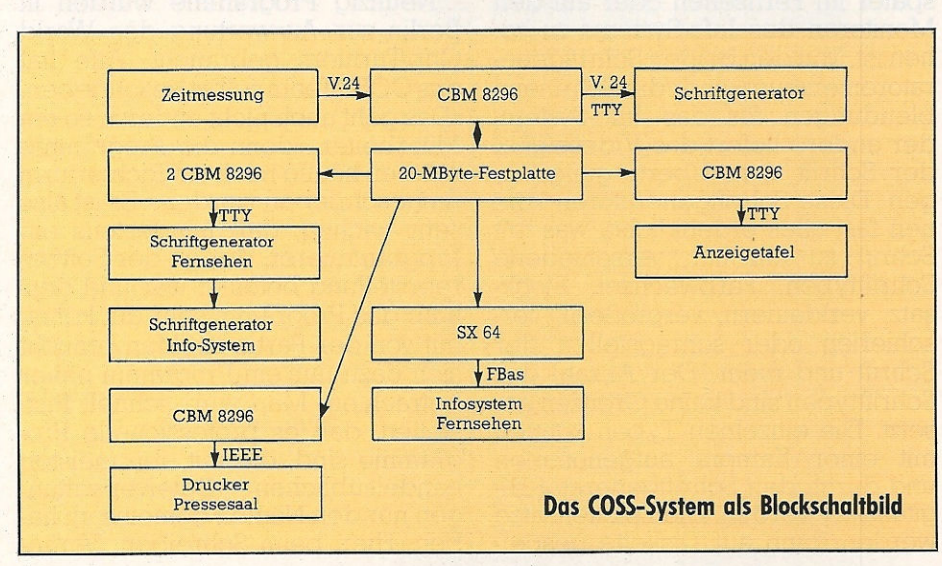
Sämtliche Datenleitungen haben einen gemeinsamen Knotenpunkt: zwei 20-MByte-Festplattenlaufwerke, die in Berlin in einem finsteren Loch »versteckt« waren. Die Verbindung der Computer mit dem Laufwerk erfolgt über eine Schnittstelle, die von den COSS-Mitarbeitern selbst entwickelt wurde. Im Gegensatz zu einer gewöhnlichen V.24-Schnittstelle, die maximal 19200 Bit/Sekunde übertragen kann, schafft die COSS-Schnittstelle um 160000 Bit/Sekunde, also etwa das 9fache. Diese Schnittstelle wird von den IEEE-Schnittstellen der Computer und Festplattenlaufwerke angesteuert.

Datenübertragung nach eigenem Konzept

Warum man überhaupt eine Schnittstellenwandlung vornimmt und die Daten seriell sendet, liegt einfach an der Zuverlässigkeit dieser Art von Datenübertragung. Parallele Übertragung von IEEE-Signalen wäre auf der benötigten Kabellänge einfach viel zu anfällig gegen Störstrahlungen. Das Problem ist nämlich, daß die Datenleitungen in den gleichen Kabelschächten untergebracht werden müssen wie die Starkstromkabel. Und die Übertragungssicherheit muß auf jeden Fall gewährleistet sein. Schließlich holt sich jeder Computer des Systems von den Festplattenlaufwerken die Daten zu einer bestimmten Auswertung oder speichert welche ab.

Auch Lösungen für kleinere Veranstaltungen

Aus der Erfahrung, die der Commodore-Sportservice bei nationalen und internationalen Großveranstaltungen gewann, wurde ein Programm für den C 64 entwickelt, das als Steckmodul erhältlich ist. Mit diesem Modul kann jeder kleinere Verein fast jede Art von Meisterschaften oder ähnliche Veranstaltungen auswerten. Was dazu nötig ist, ist ein C 64 mit Monitor, ein 1541-Diskettenlaufwerk und ein Drucker. Mit dem Steckmodul können die Daten von bis zu 400 Teilnehmer in 40 Klassen verwaltet werden. Für größere Vorhaben bietet Commodore auch andere Lösungen an. Zum Beispiel CBM 8296-Computer mit Diskettenlaufwerk, Drucker und den entsprechenden Programmen. Wem das nicht leistungsfähig genug ist, kann sich auch den Commodore-Sportservice mieten. Sollten Sie interessiert sein, erfahren Sie die Kosten auf Anfrage bei Commodore in Frankfurt. (hm)



Das COSS-System als Blockschaltbild

Fortsetzung von Seite 40

EEPROMs wurden dabei allerdings nicht berücksichtigt. Die Software erlaubt das komfortable aber leider auch falsche Programmieren. Mit dem eingebauten Monitor werden Programme eingeladen, beziehungsweise ausgelesene EPROMs abgespeichert. Alle anderen wesentlichen Funktionen erreicht man durch einfache Auswahl aus einem Menü.

Bei der Programmierzeit wurde ein Kompromiß zwischen Geschwindigkeit und Datensicherheit eingegangen. Während des Tests gab es allerdings keinerlei Schwierigkeiten mit den programmierten EPROMs. Sie enthielten alle eingegebenen Werte. Einziger Nachteil dieses ab 248 Mark teuren Gerätes ist wohl der nicht durchgeschleifte Expansion-Port. Nach dem Einsetzen des Brenners ist dieser belegt und kann nicht mehr für andere Module verwendet werden.

Von Roßmöller wurde ein wesentlich verbessertes EPROM-Programmiergerät angekündigt. Damit sollen sich auch EPROMs und EEPROMs vom Typ 27512 programmieren lassen. Weiterhin soll dieses Gerät mit einer RS232- und einer Centronics-Schnittstelle (6-KByte-Druckerpuffer) ausgestattet sein. Dies soll zusammen mit einer 6502-CPU den Anschluß an alle Computer mit dieser Schnittstelle ermöglichen. Das Brennen von Basic-Programmen mit Autostart soll ebenso wie ein wesentlich verbessertes Video-Signal das Gesamtbild abrunden. Der Einführungspreis liegt nach Angaben des Herstellers bei 398 Mark. Über Lieferzeiten war noch nichts zu erfahren.

Fazit

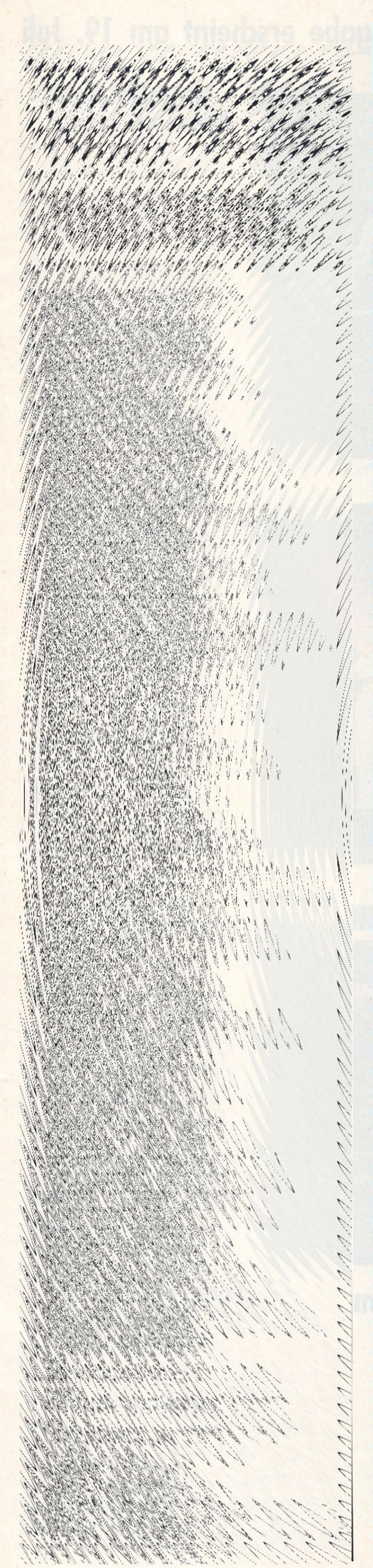
Es hat sich etwas getan. Die Hersteller von EPROM-Programmiergeräten haben erkannt, daß hier ein weites Betätigungsfeld für den Anwender liegt. Dementsprechend ausgereift sind auch die getesteten Geräte. Es war kein Gerät im Test, das nicht als gut bezeichnet werden könnte. Besonders gefallen hat der PP 64. Auch das Dela-Programmiergerät mit seinem unschlagbaren Preis kann mit zur Spitzengruppe gezählt werden. Tapfer geschlagen hat sich auch unser bisheriges Referenzgerät (das mittlerweile 198 Mark kostet), das bis auf die nicht mehr zeitgemäße Einstellung per Schalter nur vom PP 64 geschlagen wurde. Interessant dürfte auch der »Neue« von Roßmöller werden.

(Arnd Wängler/ah)





Inserentenverzeichnis





Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael M. Pauly (py)
Stellv. Chefredakteur: Michael Scharfenberger (sc)
Redakteure: aa = Albert Absmeier, leitender Redakteur, ah = Achim Hübner, ev = Volker Everts, gk = Georg Klinge, hm = Harald Meyer, rg = Christian Rogge
Redaktionsassistenz: Gerda Vogl (202)
Fotografie: Janos Feitser/Jens Jancke, Titelfoto: Jens Jancke

Layout: Leo Eder (Ltg.), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug, Tel. 042-223155/56, Telex: 862329 mut ch
USA: M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. (415) 424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenleitung: Christian Schlottau (282)

Anzeigenverwaltung und Disposition: Michaela Hörl (171)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297x210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 2 vom 1. Januar 1985.
Anzeigengrundpreise: 1/1 Seite sw: DM 8500,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1400,-. Vierfarbzuschlag DM 3800,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/2-Seite
Anzeigen im Computer-Markt: Die ermäßigten Preise im Computer-Markt gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 6400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 11,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-119. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,50. Der Abonnementspreis beträgt im Inland DM 78,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall

Verantwortlich: Für redaktionellen Teil: Michael M. Pauly. Für Anzeigen: Christian Schlottau.

Redaktions-Direktor: Michael Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon 089/4613-0, Telex 522052



Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Neuer Pascal-Compiler

Die 64'er-Redaktion hatte als erste die Möglichkeit, den neuen Pascal-Compiler von Data Becker zu testen. Handelt es sich nur um eine weitere Pascal-Version unter vielen, oder wird hier ein neuer Maßstab gesetzt? Jedenfalls sollten Sie mit der Anschaffung eines Pascal-Compilers bis zur nächsten Ausgabe des 64'er-Magazins warten — es könnte sich lohnen.

Forth-Compiler zum Abtippen

Das Listing des Monats in der Ausgabe 8 ist wieder einmal ein ganz besonderer Leckerbissen: Ein Forth-Compiler für den C 64 erwartet Sie. Aber keine Angst, es kommen keine endlosen DATA-Wüsten auf Sie zu, denn dieser Compiler ist in Basic geschrieben. Dadurch ist das Compilieren selbst zwar etwas langsam, die übersetzten Programme laufen jedoch mit der von Forth gewohnten Geschwindigkeit. Und so ganz nebenbei lernen Sie beim Abtippen des Programms die Arbeitsweise eines Forth-Compilers kennen.

Hypra-Save

Nachdem Hypra-Load bei den Lesern auf so großes Interesse gestoßen ist, werden wir in der nächsten Ausgabe das Gegenstück Hypra-Save veröffentlichen. Hypra-Save ist voll kompatibel zu Hypra-Load und ermöglicht ein erheblich schnelleres Abspeichern auf Diskette. Diese beiden Programme geben jedem Besitzer eines 1541-Diskettenlaufwerks ein Werkzeug an die Hand, das so manchen hardwaremäßigen Versuch, das langsame Diskettenlaufwerk zu beschleunigen, bei weitem übertrifft.

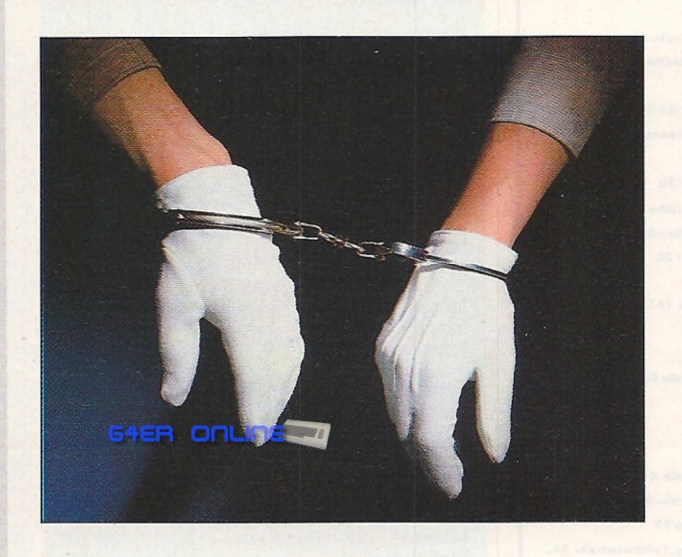
Außerdem...

- Zwei Wettbewerbe mit tollen Preisen
- Disk-Monitor
- Berichte von der CES in Chicago
- und wieder viele Tips und Tricks für den C 64, VC 20 und C 16



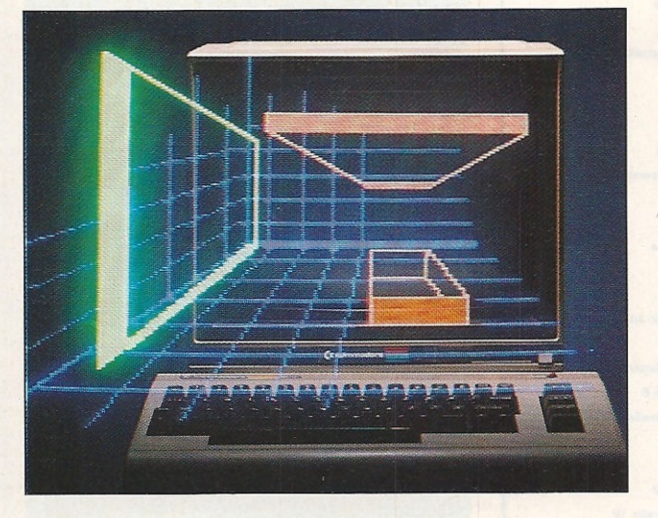
Neueste Spiele

In dieser Ausgabe begannen wir mit einer neuen Art des Spieletests. Fortgesetzt werden diese Tests unter anderem mit den Spielen »Mail Order Monsters« und »Racing Destruction Set«. Weiterhin finden Sie einen ausführlichen Bericht über das neue Spiel zum James Bond-Film.



Was macht der Sumpf?

Die Raubkopiererszene hat sich in der letzten Zeit geändert. Was ist aus den ehemaligen Crackern geworden, wie ist die Einstellung der heute noch Aktiven? Wir werden auch die rechtliche Seite des verbotenen Kopierens und Tauschens beleuchten.



Alles über Grafik

Grafik ist wohl immer noch eines der interessantesten Themen für den C 64. In einem ausführlichen Vergleichstest stellen wir Ihnen die bekanntesten Grafik- und Malprogramme vor und machen Sie mit der Funktionsweise eines Lightpens, Trackballs und Grafiktabletts vertraut.

Neuer Kurs: Commodore Extern — der Weg nach draußen

Interessiert es Sie, wie Sie über den User-Port alle möglichen Geräte steuern können? Möchten Sie sich einen Lichtgriffel für unter 5 Mark selber bauen? Oder wollen Sie wissen, wie man Meßgeräte über den Control-Port abfragt? Unser neuer Kurs führt Sie in die Geheimnisse der Hardware sowohl des C 64 als auch des VC 20 ein. Keine Angst, Sie brauchen nicht Elektronik studiert zu haben, um diesen Kurs mitverfolgen und mitmachen zu können. Anhand einfacher und nachbausicherer Bauanleitungen und kleiner Demo-Programme ist der leichte Einstieg in Theorie und Praxis des »Commodore Extern — der Weg nach draußen« garantiert — und das auch, wenn Sie sich bisher noch nicht mit Hardware-Basteleien auseinandergesetzt haben.



